

EdScratch Lektionen

Leitfaden für den Unterricht und Antwortschlüssel



Die EdScratch-Lektionen von Kat Kennewell und Jin Peng sind lizenziert unter einer [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

Inhalt

Über diesen Leitfaden	3
Was in diesem Leitfaden steht	3
Creative Commons Lizenz	4
Wie Sie diesen Leitfaden verwenden	4
Die Aktivitätsarten verstehen.....	4
Lesen der Aktivitätenübersicht	5
Verwenden des Lösungsschlüssels	7
Benötigte Materialien	8
Häufig gestellte Fragen.....	10
Bevor Sie beginnen	15
Edison vorbereiten	15
Einrichten Ihrer Geräte zur Programmierung mit EdScratch.....	17
Fehlerbehebung.....	19
Zusammenfassung der Lektionen.....	24
Lektion 1: Erste Schritte	28
Lektion 1: Edison kennenlernen	29
Lektion 2: EdScratch kennenlernen	48
Einheit 2: Los!	56
Lektion 1: Sequenz	58
Lektion 2: Eingaben und Ausgaben	70
Einheit 3: Schleifen programmieren.....	96
Lektion 1: Schleifen	98
Lesson 2: Unterbrechungen	118
Lektion 4: Was wäre, wenn...	131
Lektion 1: Bedingungen	134
Lektion 2: Fühlen	149
Einheit 5: Vielseitige Variablen.....	177
Lektion 1: Mathe und Daten in EdScratch	179
Einheit 6: Erfinder!	203
Lektion 1: Entwerfen, Bauen, Testen, Wiederholen	204
Anhang 2: Kalibrieren der Hinderniserkennung	217

Über diesen Leitfaden

Dieser Leitfaden bietet Lehrern und Ausbildern Übersichten, Unterrichtsempfehlungen und andere unterstützende Informationen für die EdScratch-Unterrichtsaktivitäten, erhältlich unter <https://meet Edison.com/robot-programming-software/edscratch/>.

Die EdScratch-Aktivitäten sind so konzipiert, dass die Schüler selbständig arbeiten können. Während die Schülerinnen und Schüler die Aktivitäten durcharbeiten, entwickeln sie eine Vertrautheit mit rechnergestütztem Denken und grundlegenden Konzepten der Informatik und erlernen dabei eine Reihe von verwandten Fähigkeiten. Dieser Leitfaden gibt einen Überblick über die zu lehrenden Konzepte, die zu verfolgenden Ansätze, die zu entwickelnden Fähigkeiten und die zu erlernenden Prozesse. Hier finden Sie Strategien, Ideen und weitere Informationen, die dazu beitragen, den Unterricht mit Edison und EdScratch effektiv, einfach und unterhaltsam zu gestalten.

Was in diesem Leitfaden steht

Die EdScratch-Unterrichtsaktivitäten sind in sechs Lektionen organisiert - von einer anfänglichen Vorbereitungseinheit bis zu einer abschließenden projektbasierten Einheit - wobei jede Lektion ein oder zwei „Unter“-Lektionen enthält. Jede Lektion enthält eine Mischung aus strukturierten und ergebnisoffenen Aktivitäten, die die Kernkompetenzen und Lernziele einführen und gleichzeitig die Schülerinnen und Schüler in eine aktive Auseinandersetzung mit dem Gelernten verwickeln.

Dieser Leitfaden bietet einen Überblick und unterstützende Informationen für alle EdScratch-Einheiten, Lektionen und Aktivitäten. In diesem Leitfaden finden Sie:

- einen Überblick über jede Lektion, einschließlich der wichtigsten Lernziele dieser Lektion,
- eine Übersicht über jede Lektion mit einer Liste aller Aktivitäten in dieser Lektion,
- ein spezieller Abschnitt pro Aktivität mit Einzelheiten:
 - Zweck und Ziele der Aktivität,
 - unterstützende Informationen, einschließlich des geschätzten Zeitbedarfs, der Anforderungen und Tipps für die Bearbeitung, und
 - einen Antwortschlüssel mit Empfehlungen zur Bewertung von Schülerarbeiten.

Creative Commons Lizenz

Diese Lehrmittel sind unter einer Creative-Commons-Lizenz veröffentlicht worden. Es steht Ihnen frei, diese Ressourcen so zu verwenden, wie sie sind, sie zu übersetzen, zu teilen oder sie als Grundlage für die Entwicklung Ihrer eigenen maßgeschneiderten Lektionen zu verwenden.

Lizenz- und Zuweisungsangaben

Das EdScratch-Lektionsplan-Set besteht aus den EdScratch-Lektionsaktivitäten und diesem Leitfaden. Die Sammlung ist unter einer [Creative Commons Attribution-ShareAlike 4.0 International License](#) lizenziert.

Entwickelt und geschrieben von: Kat Kennewell

Illustrationen von: Jin Peng

EdScratch wird von Microbric Pty Ltd unter Verwendung von Open Source Software entwickelt, die von der Scratch Foundation erstellt und gewartet wird. Die Scratch Foundation sponsert, unterstützt oder autorisiert diesen Inhalt nicht. Siehe scratch.mit.edu für weitere Informationen.

Wie Sie diesen Leitfaden verwenden

Dieser Leitfaden ist um die Aktivitäten und Lektionen herum organisiert, die das EdScratch-Lehrangebot ausmachen.

Die Aktivitäten der EdScratch-Lektion sind in sechs Lektionen unterteilt, von denen jede eine oder zwei „Unter“-Lektionen hat. Jede Lektion enthält eine Reihe von Aktivitäten, die alle mit dem Thema dieser Lektion zusammenhängen und mit den Lernzielen dieser Lektion verknüpft sind.

Die Aktivitätsarten verstehen

Die EdScratch-Lektionen bestehen aus Aktivitäten, die darauf abzielen, computergestütztes Denken, Informatik und Problemlösungsfähigkeiten schrittweise einzuführen und zu festigen. Alle Lektionen beinhalten Basisaktivitäten, die Kernkompetenzen einführen und Aufgaben enthalten, die von den Lernenden verlangen, das Gelernte anzuwenden. Ergänzende Erweiterungsaktivitäten, die dazu dienen, die Kernlektionen jeder Lektion weiter zu erforschen und zu erweitern, sind ebenfalls enthalten.

Es gibt drei Arten von Lektionsaktivitäten: Aktivitäten zu "Erkunden", Aktivitäten zu "Veränderung" und Aktivitäten zu "Herausforderung".

Lassen Sie uns die Aktivitäten untersuchen - Grundprinzipien

Alle Lektionen beinhalten "Lasst uns erkunden" Aktivitäten. Diese Aktivitäten sind die Basisaktivitäten, die neue Konzepte einführen und Kernkompetenzen erklären. Zu den "Lasst uns erkunden"-Aktivitäten gehören Erklärungen und Anleitungen rund um die Kernlektionen, dazu gibt es Aufgaben, die die Schülerinnen und Schüler erledigen müssen. Die Aufgaben ermöglichen es den Schülerinnen und Schülern, die eingeführten Konzepte und Fähigkeiten zu erkunden.

Change it up-Aktivitäten - Erweiterungsaktivitäten

„Change it up“-Aktivitäten sind Erweiterungsaktivitäten, die bereits eingeführte Fähigkeiten und Konzepte stärken. Mit den "Change it up"-Aktivitäten wird zwar kein neues Material eingeführt, aber sie bieten neue Projekte oder Aufgaben, die es auszuprobieren gilt. Die "Change it up"-Aktivitäten sind im Allgemeinen weniger strukturiert als die "Let's explore"-Aktivitäten, enthalten aber immer noch genügend Anleitung und Hinweise, um die Schülerinnen und Schüler bei der Durchführung der Aufgaben oder Projekte zu unterstützen.

Meet the Challenge-Aktivitäten – Erweiterungsaktivitäten

Ähnlich wie die "Change it up"-Aktivitäten sind auch die Meet the Challenge-Aktivitäten Erweiterungsaktivitäten, die das zuvor eingeführte Material erweitern und verstärken. Die in den Meet the Challenge-Aktivitäten umrissenen Aufgaben und Projekte sind im Allgemeinen offener und bieten weniger strukturierte Anleitung als die "Let's explore"- oder "Change it up"-Aktivitäten.

Lesen der Aktivitätenübersicht

Jede EdScratch-Lektionsaktivität hat einen eigenen Abschnitt in diesem Leitfaden. Pro Aktivität finden Sie die folgenden Informationen:

- die ID-Nummer und den Namen der Aktivität,
- eine Schätzung des Umfangs der Aktivität,
- Bearbeitungsempfehlungen (falls vorhanden),
- eine Liste der für die Aktivität benötigten Materialien und Ressourcen,
- einen Überblick über die Aktivität,
- Tipps oder Tricks, die bei der Durchführung der Aktivität helfen, und
- einen Antwortschlüssel mit Bewertungsvorschlägen (je nach Bedarf).

Ein Hinweis zu Schätzungen des Aktivitätsumfangs

Die geschätzte Größe (angegeben als "klein", "mittel", "groß" oder "Projekt") ist ein Hinweis auf den Umfang der Aktivität im Vergleich zu anderen Aktivitäten in der EdScratch-Lektion. Die für die Durchführung einer Aktivität benötigte Zeit kann je nach Alter, Erfahrung und Interesse der einzelnen Schülerinnen und Schüler an der Aktivität stark variieren.

im Allgemeinen:

- „Kleine“ Aktivitäten erfordern weniger als einen Standard-Lehrblock (weniger als 45 Minuten)
- „mittlere“ Aktivitäten sollten innerhalb eines Standard-Lehrblocks (45 Minuten oder weniger) abgeschlossen werden können
- Bei „großen“ Aktivitäten können die Schülerinnen und Schüler mehr als einen vollen Unterrichtsblock benötigen (ca. 45 Minuten - 90 Minuten).
- Projekte sind komplexe Aktivitäten, die in der Regel mehrere Unterrichtsblöcke erfordern (mindestens 90 Minuten).

Es wird empfohlen, einige einleitende Aktivitäten mit Ihren Schülerinnen und Schülern durchzuarbeiten, um abzuschätzen, wie lange sie im Durchschnitt brauchen werden, um Aktivitäten jeder Größe durchzuführen.

Verwenden des Lösungsschlüssels

Alle EdScratch-Aktivitäten haben Aufgaben oder Projekte, die abgeschlossen werden müssen. Viele der Arbeitsblätter verlangen von den Lernenden auch, Elemente ihrer Erfahrungen zu notieren oder Antworten auf Fragen in verschiedenen Formaten zu geben. Für jede Aktivität, bei der die Schülerinnen und Schüler Antworten auf eine oder mehrere Fragen geben, finden Sie im Abschnitt der jeweiligen Aktivität einen Lösungsschlüssel.

Der Lösungsschlüssel listet sowohl die Fragennummer als auch den Fragentyp auf und liefert dann die Antwort (oder Antwort/Beispielantwort) zusammen mit Markierungshinweisen, falls erforderlich.

Beispiel Lösungsschlüssel:

Frage	Typ	Antwort/Beispielantwort	Markierungsnotizen
1	SE	<i>Als ich Edison einschaltete, leuchteten die roten LEDs auf und begannen zu blinken. Edison machte auch einmal ein zwitschern-des Geräusch.</i>	Bei den Antworten der Schülerinnen und Schüler ist auf zwei wichtige Dinge zu achten: - Die roten LED-Leuchten des Roboters leuchten auf und beginnen zu blinken. - Der Roboter macht ein Geräusch.

Antwortarten

Die Spalte für den Antworttyp gibt an, welche Art von Antwort die Schülerinnen und Schüler auf die Frage geben sollten. Es gibt drei mögliche Antworttypen:

- Exakte Antwort (EA):** eine Frage, die eine genaue Lösung hat, die der Schüler/die Schülerin angeben muss.
- Ergebniscode (RC):** eine Frage, bei der die Schülerinnen und Schüler aufgefordert werden, den Code zu erfassen, den sie als Ergebnis ihrer Programmierung haben, wobei eine Vielzahl von Codelösungen möglich ist.
- Schülererfahrung (SE):** eine Frage, die die Schülerinnen und Schüler auffordert, ihre Erfahrungen oder Ergebnisse als Ergebnis ihrer Experimente und Programmierungen zu erfassen, wobei eine Vielzahl von Antworten möglich ist.

Es ist wichtig, bei der Bewertung der Arbeit der Lernenden die Art der Antwort zu beachten. Bei der Markierung von Antworten vom Typ Ergebniscode (RC) oder Schülererfahrung (SE) ist es wichtig, daran zu denken, dass viele Antworten möglich sind.

Benötigte Materialien

Alle Unterrichtsaktivitäten erfordern einen Satz von V2.0 Edison-Robotern und internetfähige Programmiergeräte (Desktops, Laptops, Chromebooks oder Tablets). Bitte lesen Sie den Abschnitt "**Bevor Sie beginnen**" in diesem Handbuch sorgfältig durch, um zu erfahren, wie Sie Ihre Programmiergeräte für die Arbeit mit EdScratch einrichten. Idealerweise werden Sie eine gleiche Anzahl von Edison-Robotern und Programmiergeräten haben wollen.

Die Mehrheit der Unterrichtsaktivitäten ist so ausgelegt, dass sie im Verhältnis 1:1 Schüler/Roboter arbeiten. Einige der Aktivitäten eignen sich jedoch am besten für Paar- oder Gruppenarbeit, bei der sich mehrere Schüler einen oder mehrere Edison-Roboter teilen. Robotik kann für einige Schülerinnen und Schüler anfangs eine überwältigende Aufgabe sein. Daher kann es oft hilfreich sein, einen unterstützenden Partner zu haben.

Die Mehrheit der Basisaktivitäten von "Let's explore" kann mit einfachen Hilfsmitteln zusammen mit den Schülerarbeitsblättern und Aktivitätsblättern, die in den EdScratch-Lektionsaktivitäten enthalten sind, durchgeführt werden. Viele der Erweiterungsaktivitäten "Change it up" und „Meet the Challenge“ erfordern zusätzliche Materialien, damit die Schülerinnen und Schüler den größtmöglichen Nutzen aus der physischen Datenverarbeitung ziehen können. Im Folgenden finden Sie eine detaillierte Auflistung der Grundausrüstung und der wichtigsten zusätzlichen Ressourcen, die Sie benötigen, um das meiste aus den EdScratch-Lektionsaktivitäten herauszuholen.

Grundausrüstung:

- Vollständiger Satz von Edison-Robotern und EdComm-Programmierkabeln
- Vollständiger Satz von Programmiergeräten (Computer oder Tablets)
- 4x AAA-Batterien pro Roboter (weitere Informationen zu Batterien finden Sie unter 'Edison bereit machen' in diesem Leitfaden)
- Ausdrücke von Schülerarbeitsblättern und Aktivitätenblättern

Hinweis: In den meisten Lektionen wird im Abschnitt "**Grundausrüstung**" der Aktivitätsübersicht „Benötigte Ausstattung“ als Punkt aufgeführt. Ein "Basissatz" bedeutet in diesem Fall ein vollständiger Satz Edison-Roboter, EdComm-Programmierkabel und Programmiergeräte sowie Batterien pro Roboter.

Zusätzliche Hilfsmittel:

- Dunkel gefärbtes Klebeband (z.B. Panzertape, schwarz)
- Taschenlampe
- Nicht durchsichtige Objekte, die als Hindernisse und Wände für 3D-Labyrinth verwendet werden
- TV- oder DVD-Fernbedienungen
- EdCreate Bausätze und alle anderen LEGO-Stein-kompatiblen Bausystemteile
- Verschiedene ‚Maker-Space‘ Handwerks- und Bastelmaterialien [wie Farbpapier, großformatiges Papier, Leim, Filz, Pappe, Pfeifenreiniger, recycelte Materialien und andere ähnliche Materialien]

Häufig gestellte Fragen

Der folgende Abschnitt enthält einige der häufigsten Fragen zur Verwendung des EdScratch-Aktivitäts-Set. Zusätzliche Informationen über die Verwendung der Edison-Roboter, einschließlich Hilfe bei der Fehlersuche, sind auf der Website Meet Edison verfügbar: www.meetedison.com

F: Wie viel müssen die Schülerinnen und Schüler bereits über Programmierung und Robotik wissen, um diese Lektionen nutzen zu können?

A: Es sind keine Vorkenntnisse in Programmierung oder Robotik erforderlich, um die EdScratch-Lektionen zu absolvieren. Wenn Sie oder Ihre Schülerinnen und Schüler neu im Bereich der Programmierung und Robotik sind, werden Sie die Lektionen in der Reihenfolge, wie sie in diesem Leitfaden vorgestellt werden, als den besten Weg erfahren, Ihr Wissen zu erweitern und das Beste aus diesen Aktivitäten herauszuholen.

F: Müssen die Schülerinnen und Schüler Scratch kennen, um diese Lektionen nutzen zu können? Sind diese Lektionen mit Scratch verknüpft?

A: In den Lektionen wird davon ausgegangen, dass die Schülerinnen und Schüler noch nie eine Scratch-Programmierungsumgebung verwendet haben. Wenn die Schülerinnen und Schüler jedoch mit Scratch vertraut sind, werden sie die Verwendung der EdScratch-Umgebung wahrscheinlich ziemlich unkompliziert finden.

Die EdScratch-Lektionen behandeln Basis-Programmierung und Konzepte, die auch mit der Scratch-Programmierungsumgebung des MIT gelehrt oder verstärkt werden können. EdScratch und Scratch sind jedoch getrennte Programmiersprachen - Programme, die in der einen Sprache geschrieben wurden, sind nicht mit der anderen kompatibel. Wenn Sie Scratch auch mit Ihren Lernenden verwenden, möchten Sie vielleicht eine Aktivität durchführen U1-2.1b : Ist EdScratch = Scratch? (Lektion 1, Lektion 2), um dieses Konzept weiter zu untersuchen.

F: Müssen die Lernenden alle Aktivitäten in einer Lektion durchführen, um die Lernziele dieser Lektion abzudecken?

A: Die EdScratch-Lektionen sind so gestaltet, dass sie nach und nach wichtige computergestützte Denk-, Informatik- und Problemlösungsfähigkeiten einführen und stärken. Alle Lektionen beinhalten 'Lasst uns erforschen' Aktivitäten. Diese Aktivitäten führen neue Konzepte und Begriffe ein und erklären Kernkompetenzen. Diese Kernlektionen werden dann durch die Erweiterungsaktivitäten "Change it up"

und „Meet the Challenge“ in der Lektion erweitert und verstärkt. Dieser modulare Aufbau soll ein flexibleres Lehren und Lernen durch eine Mischung aus Arbeiten mit der ganzen Klasse, in kleinen Gruppen, Paararbeit und Einzelarbeit ermöglichen.

Wenn die Schülerinnen und Schüler alle "Lasst uns erkunden"-Aktivitäten in einer Lektion absolvieren, wird sichergestellt, dass sie in alle wichtigen rechnergestützten Denk- und Lernziele der Lektion eingeführt werden. Möglicherweise werden Sie feststellen, dass die Verwendung von mindestens einigen der Erweiterungsaktivitäten "Change it up" oder „Meet the Challenge“ in einer Lerneinheit den Lernenden helfen wird, ihr Verständnis wichtiger Themen zu festigen und Fähigkeiten aufzubauen. Die Einbeziehung einiger der Erweiterungsaktivitäten "Change it up" und „Meet the Challenge“ stellt auch sicher, dass die Schülerinnen und Schüler reale Problemlösungs- und Robotik-Projektmöglichkeiten kennenlernen.

F: Sind die Lösungen im Lösungsschlüssel die einzigen richtigen Antworten?

A: In einigen Fällen gibt es eine einzige richtige Antwort, aber in den meisten Fällen sind viele Lösungen möglich. Programmieren ist kreativ - seien Sie nicht zu schnell dabei, die Denkweise Ihrer Schülerinnen und Schüler abzutun. Was als "falsch" erscheinen mag, kann eine akzeptable Lösung sein, auch wenn eine elegantere Lösung möglich ist.

F: Kann ich die vorgeschlagenen Aktivitäten und Projekte ändern?

A: Unbedingt! Werkzeuge, wie Edison und EdScratch, können allein schon starke Motivatoren sein. Um jedoch das Beste aus einem "coolen Werkzeug" herauszuholen, muss es sich direkt auf die Interessen der Schülerinnen und Schüler beziehen, die es benutzen. Die Auswahl von Erweiterungsaktivitäten und -projekten, die für das Leben und die Interessen Ihrer Schülerinnen und Schüler am relevantesten sind, wird ihnen helfen, das Beste aus dem Unterricht insgesamt herauszuholen.

F: Muss ich den Lektionen der Reihe nach folgen? Müssen die Lektionen und Aktivitäten innerhalb einer Lektion in der richtigen Reihenfolge durchgeführt werden?

A: Die EdScratch-Lektionen sind so gestaltet, dass sie einen flexiblen Unterricht ermöglichen. Sie können sich dafür entscheiden, den Inhalt neu zu arrangieren, um den Bedürfnissen Ihrer Lernenden am besten gerecht zu werden. Zum Beispiel können Sie einige Lektionen oder Aktivitäten verschieben, um sie besser auf andere Lernbereiche abzustimmen, in denen Ihre Lernenden zu diesem Zeitpunkt tätig sind.

Wenn Ihre Lernenden bereits Edison-Roboter benutzt haben (z.B. mit der EdBlocks-Programmierungsumgebung), können Sie auch einige Aktivitäten überspringen.

Das Arbeitsblattset für Lernende enthält auch einen Index am Ende der Sammlung. Wenn Sie nach Lektionen suchen, die sich auf ein bestimmtes Thema beziehen (z.B. Sequenz oder Algorithmen) oder die eine bestimmte Eigenschaft von Edison verwenden (z.B. die Motoren oder den Zeilenverfolgungssensor), kann Ihnen dieser Index bei der Auswahl von Aktivitäten aus verschiedenen Lektionen hilfreich sein.

Praktische Fertigkeiten in der Informatik müssen jedoch aus einigen grundlegenden Schlüsselkonzepten aufgebaut werden, und einige Elemente des Unterrichts sind progressiv. Daher ist es für Sie vielleicht am einfachsten, die Lektionen und Aktivitäten nacheinander durchzuarbeiten.

F: Sollte die ganze Klasse gemeinsam an der gleichen Aktivität arbeiten? Können die Schülerinnen und Schüler vorwärts springen?

A: Wie Sie die Inhalte vermitteln, liegt wirklich bei Ihnen! Alle Lektionen sind so gestaltet, dass Ihre Schülerinnen und Schüler selbständig lernen können. Sie können sich dafür entscheiden, die Schülerinnen und Schüler in ihrem eigenen Tempo durch alle oder einen Teil des Materials voranschreiten zu lassen. Lektionen sind gute Haltepunkte, so dass Sie die Schülerinnen und Schüler z.B. eine Lektion selbständig durcharbeiten lassen können, aber nicht weiter.

Je nach Alter, Fähigkeiten und Vertrautheit Ihrer Schülerinnen und Schüler mit Edison-Robotern und Scratch-Programmierungsumgebungen stellen Sie möglicherweise fest, dass einige Schülerinnen und Schüler Aktivitäten schneller abschließen als andere. Wenn Sie den Schülerinnen und Schülern Erweiterungsaktivitäten "Change it up" und „Meet the Challenge“ zur Verfügung stellen, die sie in ihrem eigenen Tempo durcharbeiten können, kann dies dazu beitragen, dass sich die gesamte Klasse engagiert, aber am selben Thema arbeitet.

Da es beim Programmieren im Kern um Beteiligung, Zusammenarbeit, Kreativität und Austausch geht, ist es eine weitere gute Möglichkeit, Schüler, die eine Aktivität schneller abschließen, mit anderen, die zusätzliche Zeit benötigen, zusammenarbeiten zu lassen, um die gesamte Klasse engagiert zu halten und gemeinsam Fortschritte zu erzielen.

F: Auf welchem Leseniveau müssen die Schülerinnen und Schüler lesen, um die Schülerarbeitsblätter selbständig verwenden zu können?

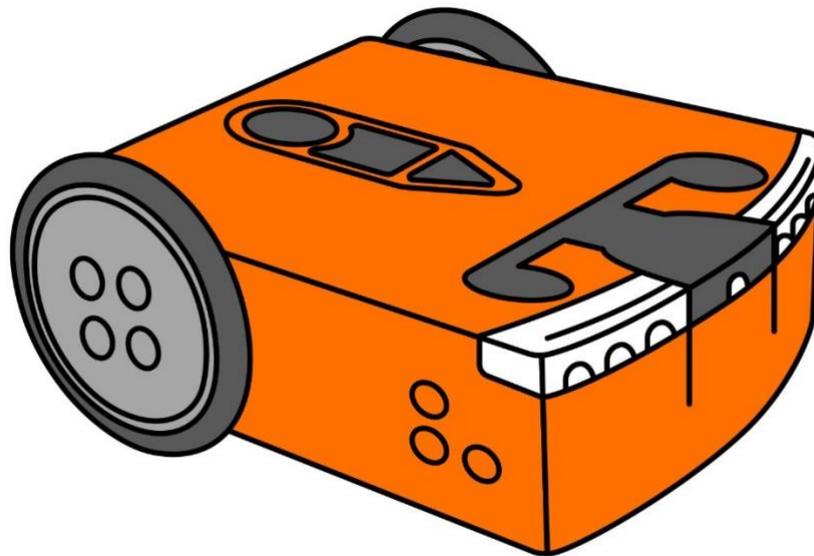
A: Der Originalsatz von Schülerarbeitsblättern, die in australischem Englisch verfasst wurden, ist für die unabhängige Verwendung durch Schüler ab Klasse 5 (10-11

Jahre) vorgesehen. Insgesamt hat der komplette Arbeitsblattsatz ein durchschnittliches Leselevel von 6,6 auf der Flesch-Kincaid Grade Level Skala, wobei spätere Lektionen im Durchschnitt etwas höhere Leseanforderungen stellen als frühere Lektionen.

Bitte beachten Sie: Übersetzte und modifizierte Versionen dieser Lektionen können unterschiedliche Leseanforderungen haben.

F: Ich habe Edison-Roboter der Version 1. Kann ich EdScratch und diese Lektionen verwenden?

A: In der Regel können Sie Edison-Roboter der Version 1, die über die neueste Firmware-Aktualisierung (verfügbar unter <https://meetiedison.com/edison-robot-support/firmware-update/>) verfügen, mit EdScratch und den Aktivitäten in diesem EdScratch-Lektionssatz verwenden. Es gibt jedoch einige Einschränkungen, besonders wenn es um die Motorleistungen von Edison geht. Wenn Sie EdScratch-Blöcke aus der Kategorie 'Antrieb' in EdScratch mit Robotern der Version 1 verwenden, verwenden Sie nur 'Sekunden' als Eingabeparameter für die



Entfernungs-Lektionen. Die Verwendung von 'cm', 'Zoll' oder 'Grad' als Eingabeparameter für die Entfernungseinheiten funktioniert nicht mit Robotern der Version 1 und kann zu einem seltsamen Verhalten des Roboters führen. Bei allen Aktivitäten und Projekten, bei denen die Schülerinnen und Schüler Blöcke der Kategorie "Antrieb" verwenden, sollten Sie die Schülerinnen und Schüler daran erinnern, immer und ausschließlich "Sekunden" als Eingabeparameter für die Entfernungseinheiten zu verwenden.

Es wird auch empfohlen, die Verwendung der Abtast- und Ereignis-EdScratch-Blöcke, die sich auf "Antriebsbeanspruchung" beziehen, bei Robotern der Version 1 zu vermeiden, da diese Blöcke dazu führen können, dass sich die Roboter unregelmäßig verhalten. (Hinweis: Keine der Aktivitäten in diesen Lektionen verwendet die Abtast- oder Ereignisblöcke zur 'Antriebsbelastung').

Was ist der Grund für diese Einschränkungen? Die Hauptursache hat mit einigen physikalischen Unterschieden zwischen Edison-Robotern der Version 1 und Edison-Robotern der Version 2.0 zu tun. Edison-Roboter der Version V2.0 verfügen über Radcodierer, die es den Robotern ermöglichen, bestimmte Strecken mit exakter Geschwindigkeit zurückzulegen. Diese Encoder ermöglichen es den Robotern, andere Entfernungseinheiten als die Zeit zu verwenden, einschließlich Zentimeter, Zoll und Grad. Version 1 Edison-Roboter der Version 1 haben keine Encoder und sind daher nicht in der Lage, die für die Verwendung dieser anderen Eingaben erforderliche Präzision zu fahren.

Gefallen Ihnen diese Lektionen oder eher nicht? Haben Sie eine Idee für eine Unterrichtsaktivität?

Das Team hinter EdScratch und Edison würde sich freuen, von Ihnen zu hören! Sie können uns Ihr Feedback und Ihre Ideen über das Kontaktformular auf unserer Website unter www.meetedison.com/edison-robot-support/contact-us/ mitteilen.

Bevor Sie beginnen

Bevor Sie Edison mit Ihren Lernenden verwenden, müssen Sie die Programmiergeräte (d.h. die Computer, Laptops, Chromebooks oder Tablets), die Sie mit der EdScratch-Anwendung verwenden werden, einrichten und die Edison-Roboter bereit machen.

Dieser Leitfaden zeigt die Grundlagen und die Hilfe bei der Fehlerbehebung für die Einrichtung zur Verwendung von Edison mit EdScratch. Zusätzliche Details, einschließlich Informationen über Edisons andere Programmiersprachen, finden Sie in der kostenlosen Anleitung Erste Schritte mit Edison, die unter <https://meetedison.com/content/Get-started-with-Edison-guide-English.pdf>

Edison vorbereiten

Um Edison einsatzbereit zu machen, müssen Sie das tun:

1. Öffnen Sie das Batteriefach auf der Rückseite von Edison und entfernen Sie das EdComm-Programmierkabel.
2. Legen Sie 4 'AAA'-Batterien ein. Bitte beachten Sie die Abbildung um sicherzustellen, dass die Batterien korrekt eingelegt sind. Achten Sie darauf, das Batteriefach wieder zu verschließen, indem Sie den Batteriefachdeckel wieder befestigen.



Vergewissern Sie sich, dass die Batterien richtig eingesetzt sind.

Bitte beachten: Schwache oder leere Batterien können eine Reihe von Problemen mit Edison verursachen. Verwenden Sie daher immer frische, voll aufgeladene Batterien in Ihren Robotern.

Wahl der Batterien: Wenn Sie Einwegbatterien mit Edison verwenden, verwenden Sie immer nur Alkalibatterien. (Dies sind die gängigsten Standard-AAA-Batterien, die Sie in fast jedem Geschäft finden). Wenn Sie wiederaufladbare Batterien mit Edison verwenden, verwenden Sie nur wiederaufladbare Nickel-Metallhydrid-Batterien (NiMH). Verwenden Sie niemals wiederaufladbare Lithium-Akkus, Hochleistungs-Einwegbatterien, Super-Hochleistungs-Einwegbatterien oder Kohlenstoff-Zink-Batterien.

3. Um Edison einzuschalten, drehen Sie den Roboter um. Schieben Sie den Netzschalter in die "Ein"-Position, wie in



Schieben Sie den Schalter in Richtung des "Ein"-Symbols.

der Abbildung gezeigt. Dadurch wird Edison eingeschaltet, und die roten LED-Leuchten beginnen zu blinken.

Bitte beachten: Obwohl Edison sich automatisch abschaltet, wenn er nach fünf Minuten nicht benutzt wird, empfehlen wir Ihnen, die Roboter manuell abzuschalten, wenn sie nicht benutzt werden.

Einrichten Ihrer Geräte zur Programmierung mit EdScratch

Der beste Weg, Ihre Programmiergeräte einzurichten, ist die Ausführung eines Testprogramms in EdScratch. Folgen Sie diesen sechs Schritten, um EdScratch auf Ihrem Gerät zu testen:

1. Laden Sie die EdScratch-Anwendung, indem Sie www.edscratchapp.com in einem Browser öffnen (wir empfehlen dringend Google Chrome). Starten Sie die Programmieranwendung, indem Sie die orangefarbene Schaltfläche 'Launch EdScratch' drücken. Stellen Sie sicher, dass Sie Pop-ups für www.edscratchapp.com zulassen.
2. Sobald sich die Anwendung öffnet, sehen Sie die Programmierumgebung. Öffnen Sie 'Menü' aus der Menüleiste und wählen Sie 'Demos laden'. Eine Liste von Demoprogrammen wird in einem Pop-up-Fenster geöffnet. Wählen Sie das Programm mit dem Namen 'Test_program', das in die Programmierumgebung geladen wird.
3. Stellen Sie die Lautstärke Ihres Gerätes auf Maximum oder 100% ein. Stecken Sie das EdComm-Programmierkabel in die Audiobuchse Ihres Geräts.

HINWEIS: Viele Geräte verfügen über eingebaute Sicherheitseinstellungen, die die Lautstärke reduzieren, wenn ein Audiogerät an die Kopfhörerbuchse angeschlossen wird. Überprüfen Sie die Lautstärkeinstellungen immer doppelt, nachdem Sie das EdComm-Kabel an Ihr Gerät angeschlossen haben.

4. Schalten Sie Ihren Edison-Roboter ein. Schließen Sie das EdComm-Kabel an der Unterseite des Roboters in der Nähe des Netzschalters an. Drücken Sie einmal die runde (Aufnahme-)Taste.
5. Drücken Sie in der EdScratch-Anwendung die Schaltfläche 'Program Edison'. Folgen Sie den Anweisungen auf dem Pop-up und drücken Sie dann die 'Program Edison'-Taste auf dem Pop-up, um das Programm in Edison herunterzuladen.

HINWEIS: Wenn die Warnmeldung "There seems to be a network issue accessing the compiler" an diesem Punkt erscheint, siehe Abschnitt "Fehlerbehebung 1: „Überprüfen Sie den Verbindungsstatus" weiter unten.

6. Während des Herunterladens des Programms hören Sie ein surrendes Geräusch, ein bisschen wie bei einem Wählmodem. Wenn der Download abgeschlossen ist, hören Sie eines von zwei Geräuschen: das "Success"-Geräusch (der gleiche zirpende Piepton, den Edison macht, wenn Sie den

Roboter zum ersten Mal einschalten) oder das "Fail"-Geräusch (ein absteigender Piepton)¹.

ERFOLGREICHE VERBINDUNG: Wenn der Roboter das Geräusch "Erfolg" macht, ziehen Sie ihn vom EdComm-Kabel ab und drücken Sie dann einmal die Dreieckstaste (Abspielen) auf Edison, um das Programm auszuführen. Wenn das Programm erfolgreich im Edison-Roboter läuft, ist Ihr Programmiergerät einsatzbereit! Sie brauchen nichts weiter zu tun, um Ihr Gerät einzurichten.

FAIL: Wenn der Roboter das Programm nicht herunterladen kann oder das Programm nicht im Roboter abgespielt wird, beachten Sie den folgenden Abschnitt "Fehlerbehebung" durch.

¹ Sie können Aufnahmen sowohl der Erfolgs- als auch der Misserfolgsklänge hören unter <https://meet Edison.com/edison-robot-support/trouble-shooting/#success-fail-sounds>

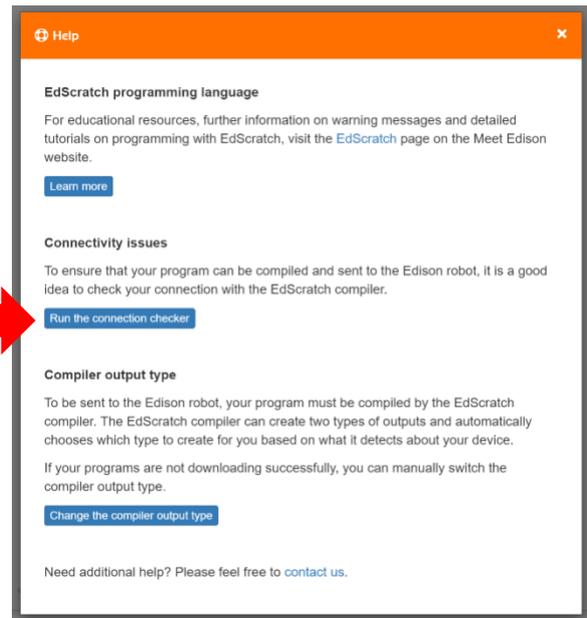
Fehlerbehebung

Abhängig von der Art der Programmiergeräte, die Sie verwenden und Ihrem Netzwerk, gibt es einige Dinge, die Sie tun müssen, um Fehler in Ihren Geräten zu beheben und sie mit der EdScratch Anwendung zum Laufen zu bringen.

Fehlerbehebung 1: Überprüfen Sie den Verbindungsstatus

Wenn Sie die Warnmeldung "There seems to be a network issue accessing the compiler" sehen, nachdem Sie die Schaltfläche "Program Edison" in der EdScratch-Anwendung gedrückt haben, oder wenn das Programm nicht erfolgreich heruntergeladen werden konnte, müssen Sie den Verbindungsstatus der Anwendung überprüfen.

Damit es funktioniert, muss die EdScratch-Anwendung auf den Compiler zugreifen (der die EdScratch-Programme in ein Format umwandelt, das an den Edison-Roboter gesendet werden kann). Innerhalb der EdScratch-Anwendung unter www.edscratchapp.com, öffnen Sie 'Menü' in der oberen linken Ecke und wählen Sie 'Help'. Dies wird ein Pop-up öffnen, welches die Option 'Run the connection checker' enthält. Klicken Sie auf diese Schaltfläche, um Ihre Verbindung zu überprüfen.



Wenn das Ergebnis des Verbindungstests 'NO SERVER FOUND' anzeigt, befinden Sie sich möglicherweise hinter einer Firewall, wie sie in Schulen üblich ist und die den Zugriff auf den Compiler blockiert. Sie benötigen den Netzwerkadministrator, um die Ports 80, 8080, 443 und 8443 freizugeben und diese Adressen auf die Weiße Liste zu setzen:

- <https://www.edscratchapp.com>
- <https://api.edisonrobotics.net>
- 52.8.213.196
- 13.210.175.93

□ 52.79.71.19

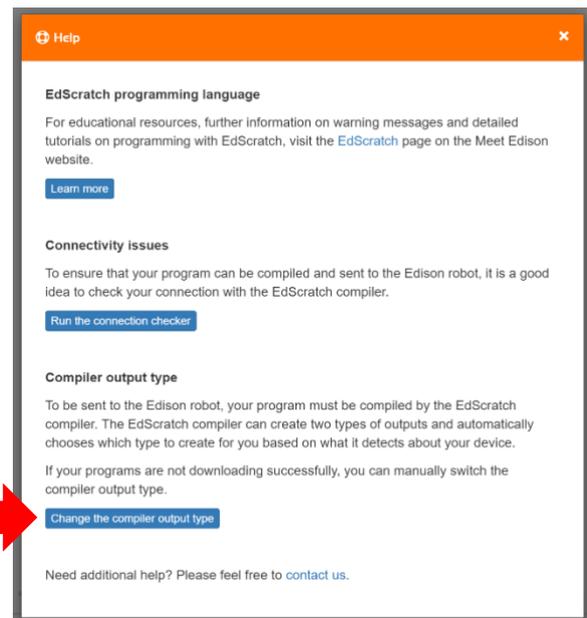
ERFOLGREICHE VERBINDUNG: Sobald der Verbindungsprüfer anzeigt, dass Sie verbunden sind, versuchen Sie erneut, ein Testprogramm herunterzuladen und auszuführen. Wenn das Programm heruntergeladen und erfolgreich im Edison-Roboter ausgeführt wird, ist Ihr Programmiergerät einsatzbereit! Sie brauchen nichts weiter zu tun, um Ihr Gerät einzurichten.

FAIL: Wenn der Verbindungsprüfer anzeigt, dass Sie verbunden sind, aber immer noch nicht in der Lage sind, Edison zu programmieren, fahren Sie mit 'Fehlerbehebung 2: Wechseln Sie den Compiler-Ausgabetyt' fort.

Fehlerbehebung 2: Umschalten des Compiler-Ausgabetyps

Um an den Edison-Roboter gesendet zu werden, muss Ihr Programm durch den EdScratch-Compiler kompiliert werden. Der EdScratch-Compiler kann zwei Arten von Ausgaben erstellen und wählt automatisch den für Sie zu erstellenden Typ, basierend auf der Art des Gerätes, das er erkennt, dass Sie es benutzen (wie z.B. ein Apple-Tablet oder einen Windows Laptop).

Wenn Ihre Programme nicht erfolgreich heruntergeladen werden, können Sie den Compiler-Ausgabetyt manuell umschalten. Innerhalb der EdScratch-Anwendung unter www.edscratchapp.com, öffnen Sie 'Menü' in der oberen linken Ecke und wählen Sie 'Help'. Daraufhin öffnet sich ein Pop-up-Fenster mit der Option "**Change the compiler output typ**". Klicken Sie auf diese Schaltfläche, um zu überprüfen, welches Gerät und welche Einstellungen erkannt werden.



Wenn das erkannte Gerät nicht genau ist oder wenn Ihre Programme nicht erfolgreich heruntergeladen werden, können Sie den Compiler-Ausgabetyt manuell umschalten. Verwenden Sie die folgenden Informationen, um die für Ihr Gerät am besten geeignete Ausgabe auszuwählen:

Compiler-Ausgang für lange Impulse

Dieser Ausgabetyt funktioniert gut auf Geräten mit geringem Ausgabevolumen, einschließlich einiger Tablets. Wenn Sie einen Mac-Computer, ein iPad-Tablet oder ein Windows- oder Android-Tablet verwenden, sollte der Langimpuls-Compilerausgang für Ihr Gerät am besten funktionieren.

Compiler-Ausgang für kurze Impulse

Dieser Ausgabebetyp funktioniert gut auf Geräten mit Tonverbesserungssoftware, einschließlich der meisten Windows-Desktop- und Laptop-Computer. Wenn Sie einen Windows-Desktop- oder Laptop-Computer verwenden, sollte die Kurzpuls-Compilerausgabe für Ihr Gerät am besten funktionieren.

ERFOLGREICHE VERBINDUNG: Wenn Sie die Compilerausgabe geändert haben, versuchen Sie erneut, ein Testprogramm herunterzuladen und auszuführen. Wenn das Programm heruntergeladen und erfolgreich im Edison-Roboter ausgeführt wird, ist Ihr Programmiergerät einsatzbereit! Sie brauchen nichts weiter zu tun, um Ihr Gerät einzurichten.

FAIL: Wenn Sie Edison nach dem Ändern des Compiler-Ausgabetyps immer noch nicht programmieren können, lesen Sie die folgenden gerätespezifischen Hinweise zur Fehlerbehebung.

Windows-Computer - Fehlerbehebung

Wenn Sie einen Laptop oder Desktop mit einem Windows-Betriebssystem betreiben und Edison nach Ausführung der obigen Einrichtungsschritte immer noch nicht programmieren können, versuchen Sie diese zusätzlichen Schritte zur Fehlerbehebung.

Ton-Verbesserungen abschalten

Wenn Sie Desktops oder Laptops mit Windows-Betriebssystemen verwenden und sowohl die Compiler-Ausgabebetypn "kurzer Impuls" (empfohlen) als auch "langer Impuls" versagen, müssen Sie die Tonverbesserungen deaktivieren.

Unter <https://meet Edison.com/edison-robot-support/trouble-shooting/#soundenhancements> finden Sie Schritt-für-Schritt-Videoanleitungen, die Ihnen zeigen, wie Sie Soundverbesserungen für Standard-Software zur Soundverbesserung von Windows sowie für die gängigsten Softwareprogramme von Drittanbietern deaktivieren können.

Sobald Sie die Klangverbesserungen deaktiviert haben, verwenden Sie die Compiler-Ausgabeoption für lange Impulse.

Prüfen Sie, ob ein Volume 'hard lock' vorhanden ist.

Einige Geräte, insbesondere in Europa, haben eine strenge Sperre für die Lautstärke, wenn ein Audiogerät erkannt wird. Das bedeutet, dass das Gerät so "gesperrt" wird, dass es bei Erkennung eines Audiogeräts nur eine maximale Lautstärke von etwa 75% der maximalen Lautstärke des Geräts liefert. Um dies zu korrigieren, gehen Sie in die Einstellungen des Geräts und deaktivieren Sie die Sperre, damit das Gerät auch bei angeschlossenem Audiogerät wirklich die volle Lautstärke abgeben kann.

Chromebooks - Fehlerbehebung

Wenn Sie ein Chromebook verwenden und Edison nach Ausführung der obigen Einrichtungsschritte immer noch nicht programmieren können, versuchen Sie diesen zusätzlichen Schritt zur Fehlerbehebung.

Tonverbesserungen deaktivieren

Einige Chromebooks, darunter auch einige Dell Chromebooks, haben eine niedrige Audioausgabe, verfügen aber auch über Klangverbesserungen. Wenn Sie ein Chromebook verwenden und sowohl der Compiler-Ausgabebetyp kurzer Impuls (empfohlen) als auch der Compiler-Ausgabebetyp langer Impuls fehlschlagen, müssen Sie die Tonverbesserungen deaktivieren.

Soundverbesserungen sind auf Windows-Rechnern üblich, und wir haben unter <https://meetedison.com/edison-robot-support/trouble-shooting/#soundenhancements> Schritt-für-Schritt-Videoanleitungen, die Ihnen zeigen, wie Sie Soundverbesserungen für Standard-Soundverbesserungen von Windows sowie für die gängigsten Softwareprogramme von Drittanbietern deaktivieren können. Je nach Hersteller verfügt Ihr Chromebook möglicherweise über eine ähnliche Software zur Klangverbesserung.

Sobald Sie die Klangverbesserungen deaktiviert haben, verwenden Sie die Compiler-Ausgabeoption für lange Impulse.

Mac-Computer - Fehlerbehebung

Wenn Sie mit einem Mac-Laptop oder -Desktop arbeiten und Edison nach Ausführung der obigen Einrichtungsschritte immer noch nicht programmieren können, versuchen Sie diesen zusätzlichen Schritt zur Fehlerbehebung.

Überprüfen Sie die Lautstärkeinstellungen

Bei einigen Macs treten Audio-Clipping-Fehler auf, wenn versucht wird, Edison zu programmieren. Wenn diese Probleme auftreten, versuchen Sie bitte stattdessen, die Lautstärke von 100% auf zwischen 50% und 90% zu reduzieren.

Tablet - Fehlerbehebung

Wenn Sie ein Apple-, Android- oder Windows-Tablet verwenden und Edison nach Ausführen der obigen Einrichtungsschritte immer noch nicht programmieren können, versuchen Sie diesen zusätzlichen Fehlerbehebungsschritt.

Überprüfen Sie die Lautstärkeinstellungen

Viele Geräte verfügen über eingebaute Sicherheitseinstellungen, die die Lautstärke reduzieren, wenn ein Audiogerät über den Kopfhöreranschluss angeschlossen wird. Bitte überprüfen Sie, ob die Lautstärke auf 100% eingestellt ist, nachdem Sie das EdComm-Programmierkabel an Ihr Gerät angeschlossen haben.

Einige Geräte, insbesondere in Europa, haben eine strenge Sperre für die Lautstärke, wenn ein Audiogerät erkannt wird. Das bedeutet, dass das Gerät so "gesperrt" wird, dass es bei Erkennung eines Audiogeräts nur eine maximale Lautstärke von etwa 75% der maximalen Lautstärke des Geräts liefert. Um dies zu korrigieren, gehen Sie in die Einstellungen des Geräts und deaktivieren Sie die Sperre, damit das Gerät auch bei angeschlossenem Audiogerät wirklich die volle Lautstärke abgeben kann.

Bitte beachten Sie: Die meisten Mobiltelefone verfügen nicht über den Audioausgang, um Edison mit EdScratch zu programmieren. Wir raten davon ab, Mobiltelefone als Programmiergeräte mit Edison zu verwenden.

Funktioniert noch nicht?

Weitere Hinweise zur Fehlerbehebung finden Sie auf unserer Website unter <https://meet Edison.com/edison-robot-support/trouble-shooting/> oder Sie können unter <https://meet Edison.com/edison-robot-support/contact-us/> kontaktieren.

Unser Team von freundlichen Mitarbeitern des Technischen Supports wird Ihnen gerne helfen!

Zusammenfassung der Lektionen

Es folgt eine Zusammenfassung jeder Lektion, einschließlich der Lernziele dieser Lektionen.

Lektion 1: Erste Schritte

Bereiten Sie sich auf Ihr EdVenture vor! Diese erste Lektion, die zwei Unterlektionen mit insgesamt vier Basisaktivitäten und zehn Erweiterungsaktivitäten umfasst, führt die Schüler zunächst in den Edison-Roboter und dann in die EdScratch-Programmierung ein. Die Schülerinnen und Schüler machen sich mit dem Edison-Roboter durch eine Reihe von Aktivitäten vertraut, einschließlich der Programmierung der Roboter mit Hilfe von Strichcodes. Anschließend lernen sie die Grundlagen der EdScratch-Programmierung als Vorbereitung auf die nächste Lektion.

Die Lernziele für diese Lerneinheit sind, dass die Schülerinnen und Schüler:

- eine Arbeitsvertrautheit mit dem Edison-Roboter entwickeln, einschließlich seiner wichtigsten physischen Merkmale (Netzschalter, Sensoren und Knöpfe),
- in der Lage sind, einen Edison-Roboter unter Verwendung von Strichcodes zu programmieren, und
- wissen, wie man auf EdScratch zugreift und wie man ein Programm von EdScratch auf den Edison-Roboter herunterlädt.

Lektion 2: Vorwärts!

Konzentrieren Sie sich in dieser Lektion, die zwei Unterlektionen mit insgesamt acht Basisaktivitäten und 13 Erweiterungsaktivitäten umfasst, auf das zentrale rechnergestützte Konzept der Sequenz. Die Schülerinnen und Schüler erforschen Edisons Fähigkeiten, sich mit Hilfe seiner Motoren zu bewegen, und nutzen die LEDs und den geräuschproduzierenden Summer des Roboters bei einer Reihe von Aktivitäten. Die Grundlagen der Computerprogrammierung einschließlich Ein- und Ausgabe, Fehlersuche und Fehlerbehebung werden vorgestellt. Die Lernenden beginnen, ihre Vertrautheit mit der Programmierung von Edison in EdScratch und mit der Verwendung von Edison als Basis für kreative Roboter-Builds zu entwickeln.

Die Lernziele für diese Lerneinheit sind, dass die Schülerinnen und Schüler

- in die Idee des rechnergestützten Denkens eingeführt werden und beginnen, rechnergestützte Rahmenwerke bei der Annäherung an Aufgaben zu verwenden,

- sich mit dem Begriff der Sequenz vertraut machen,
- lernen, was Inputs, Outputs und Eingabeparameter sind,
- EdScratch's Hauptausgabeblocke mit Edison zu erforschen, einschließlich der Kategorien 'Drive', 'LEDs' und 'Sound' in EdScratch,
- in der Lage sind, sequentielle Programme für Edison in EdScratch unter Verwendung von grundlegenden Ausgabeblocken und den Eingabeparametern dieser Blöcke zu erstellen und zu modifizieren, und
- beginnen, durch Projekte Robotikanwendungen in realen Situationen zu erforschen.

Lektion 3: Schleifen

Untersuchen Sie das wichtige Konzept der Schleife sowie verschiedene Möglichkeiten, wie Schleifen zur Steuerung des Verhaltens der Edison-Roboter in dieser Lektion, die aus zwei Unterlektionen mit insgesamt sechs Basisaktivitäten und 14 Erweiterungsaktivitäten besteht, verwendet werden können. Das Thema der Programmierlogik wird genauer untersucht, einschließlich der Frage, wie Kontrollstrukturen den Codefluss beeinflussen können. Die Schülerinnen und Schüler entdecken weiterhin die EdScratch-Umgebung, lernen mehr über die verschiedenen Blocktypen in EdScratch und die Art und Weise, wie diese Blöcke zur Erstellung mit Edison verwendet werden können.

Die Lernziele für diese Lerneinheit sind, dass die Schülerinnen und Schüler

- durch die Struktur der Computerprogrammierung von Schleifen in das Konzept des rechnergestützten Denkens der Wiederholung eingeführt werden,
- ein Arbeitsverständnis des Unterschieds zwischen bestimmten und unbestimmten Schleifen entwickeln,
- die Konzepte von Sequenz und Schleifen in EdScratch zu Arbeitsprogrammen kombinieren,
- in das Programmierkonzept der Interrupts eingeführt werden,
- mit neuen Blöcken in den Kategorien 'Kontrolle', 'Kommentare' und 'Ereignisse' in EdScratch experimentieren, und
- ihre Problemlösungsfähigkeiten durch die Erstellung und Programmierung von Robotik-Projekten erweitern.

Lektion 4: Was wäre, wenn...

Erkunden Sie Auswahl und Verzweigung in der Computerprogrammierung durch die wichtigsten rechnergestützten Konzepte von Bedingungen und Ereignissen in den zwei Unterlektionen dieser Lektion mit insgesamt neun Basisaktivitäten und 15 Erweiterungsaktivitäten. Wichtige Computerprogrammierfähigkeiten wie die Entwicklung von Pseudocode werden vorgestellt, um den Schülern zu helfen, ihre Problemlösungsfähigkeiten zu verbessern, während sie die verschiedenen Sensorfähigkeiten der Edison-Roboter freischalten. Die Schüler lernen Algorithmen kennen und nutzen dieses Verständnis, um Programme zu erstellen, die ein autonomeres Verhalten der Roboter ermöglichen. Bedingungen, Sensorik, Unterbrechungen und ereignisbasiertes Programmieren werden durch die physikalischen Rechenaktivitäten in dieser Lektion zum Leben erweckt.

Die Lernziele für diese Lerneinheit sind, dass die Lernenden

- in das Konzept des rechnergestützten Denkens der bedingten Selektion (d.h. Verzweigung) eingeführt werden,
- zu erforschen, wie Eingaben mit Edison funktionieren, indem neue Blöcke in den Kategorien 'Kontrolle', 'Empfinden' und 'Ereignisse' in EdScratch verwendet werden,
- unter Verwendung von Edisons Sensoren und Eingabemöglichkeiten in Arbeitsprogrammen experimentieren,
- die Konzepte von Sequenz, Schleifen und Auswahl in EdScratch zu Arbeitsprogrammen kombinieren,
- ein praktisches Verständnis dafür entwickeln, wie Pseudocode und Kommentare zur Planung, Verfolgung und Fehlersuche in Programmen verwendet werden können,
- in die Idee von Algorithmen in der Computerprogrammierung eingeführt werden und lernen, wie sich Algorithmen von Programmen unterscheiden, und
- ihr Verständnis von Roboteranwendungen durch Projekte erweitern, bei denen Sensorik und Inputs eingesetzt werden.

Lektion 5: Vielseitige Variablen

Tauchen Sie in die wichtigsten Berechnungskonzepte von Variablen, Daten und Ausdrücken ein und wenden Sie dabei Erlerntes aus früheren Lektionen an. Die eine Unterlektion dieser Lektion, die insgesamt vier Basisaktivitäten und acht Erweiterungsaktivitäten umfasst, rundet die Erkundung der EdScratch-Umgebung durch die Schüler ab. Frühere Konzepte werden überdacht und erweitert, um die

zusätzliche Flexibilität zu nutzen, die durch die Einbeziehung von Variablen und Operatoren zur Verwaltung von Daten innerhalb ihrer Programme geboten wird.

Die Lernziele für diese Lerneinheit sind, dass die Schülerinnen und Schüler

- in die Informatik-Grundlagen von Variablen und Daten eingeführt werden,
- untersuchen, wie Daten und Variablen mit Edison unter Verwendung neuer Blöcke in den Kategorien "Daten" und "Operatoren" in EdScratch verwendet werden können,
- die Sensoren und Eingabefähigkeiten von Edison unter Verwendung von Variablen und Operatoren zur Verfeinerung und Modifizierung des Verhaltens in Arbeitsprogrammen weiter zu erforschen, und
- die Anwendung von Informatik-Grundlagen wie Tracing und Debugging, um Projekte unter Verwendung von Abtastung und Variablen abzarbeiten.

Lektion 6: Zeit des Erfinders!

Setzen Sie Ihr ganzes Edison- und EdScratch-Wissen in die Tat um! Diese Einheit umfasst eine Lektion mit insgesamt zwei Basisaktivitäten und fünf Erweiterungsaktivitäten mit Schwerpunkt auf Projekten. Durch das Entwerfen und Entwickeln eigener Projekte unter Verwendung iterativer Planungs-, Herstellungs- und Testzyklen setzen die Schülerinnen und Schüler die wichtigsten Konzepte des rechnergestützten Denkens, Problemlösens, Programmierens und physikalischen Rechnens, die sie gelernt haben, in dieser Abschlusseinheit in die Tat um.

Die Lernziele für diese Lerneinheit sind, dass die Schülerinnen und Schüler

- etwas über den Zyklus von Entwurf, Aufbau und Test und Strategien, wie Zerlegung und iteratives Testen, für die Problemlösung im Bereich Physical Computing zu lernen
- ihr Verständnis für wichtige computergestützte Denkweisen und Prinzipien der Informatik in offenen Projekten unter Beweis stellen.

Lektion 1: Erste Schritte

Bereiten Sie sich auf Ihr EdVenture vor! Diese erste Einheit führt die Schüler zunächst in den Edison-Roboter und dann in die EdScratch-Programmierungsumgebung ein. Die Schülerinnen und Schüler machen sich mit dem Edison-Roboter durch eine Reihe von Aktivitäten vertraut, einschließlich der Programmierung der Roboter mit Hilfe von Strichcodes. Dann lernen sie die Grundlagen der EdScratch-Programmierungsumgebung als Vorbereitung für die nächste Einheit.

Lernziele

Die Schülerinnen und Schüler werden:

- eine Arbeitsvertrautheit mit dem Edison-Roboter, einschließlich seiner wichtigsten physischen Merkmale (Netzschalter, Sensoren und Knöpfe) entwickeln
- in der Lage sein, einen Edison-Roboter mit Strichcodes zu programmieren
- wissen, wie man auf EdScratch zugreift und wie man ein Programm von EdScratch auf Edison herunterlädt

Kernkompetenzen: Hardware versus Software, Edisons Knöpfe und Sensoren, EdScratch-Terminologie und grundlegende Funktionen

Lektionen und Aktivitäten in dieser Einheit

Diese Einheit umfasst zwei Lektionen mit insgesamt vier Basisaktivitäten und zehn Erweiterungsaktivitäten.

Lektion 1: Edison treffen

- [U1-1.1 Erforschen wir unsere Edison-Roboter](#)
 - o [U1-1.1a : Ziegel, Blöcke und Edison](#)
- [U1-1.2 Lassen Sie uns die Strichcode-Programmierung untersuchen](#)
 - o [U1-1.2a : Sumo-Ringen](#)
 - o [U1-1.2b : Machen Sie Ihren eigenen Strichcode?](#)
 - o [U1-1.2c Ändern Sie ihn nach oben: Strichcodes für TV-Fernbedienung](#)
 - o [U1-1.2d : Edison-Fußball](#)
 - o [U1-1.2e Herausforderung angenommen: Aufbau und Kontrolle des EdTank](#)
 - o [U1-1.2f Herausforderung angenommen: Bau und Steuerung des EdDigger](#)

- o [U1-1,2g Herausforderung angenommen: Bau und Kontrolle des EdRoboClaw](#)

Lektion 2: EdScratch kennenlernen

- [U1-2.1 Lassen Sie uns die EdScratch-Umgebung erkunden](#)
 - o [U1-2.1a Herausforderung angenommen: Laden Sie eine weitere herunter!](#)
 - o [U1-2.1b : Ist EdScratch = Scratch?](#)
- [U1-2.2 Wir erforschen Warnmeldungen](#)

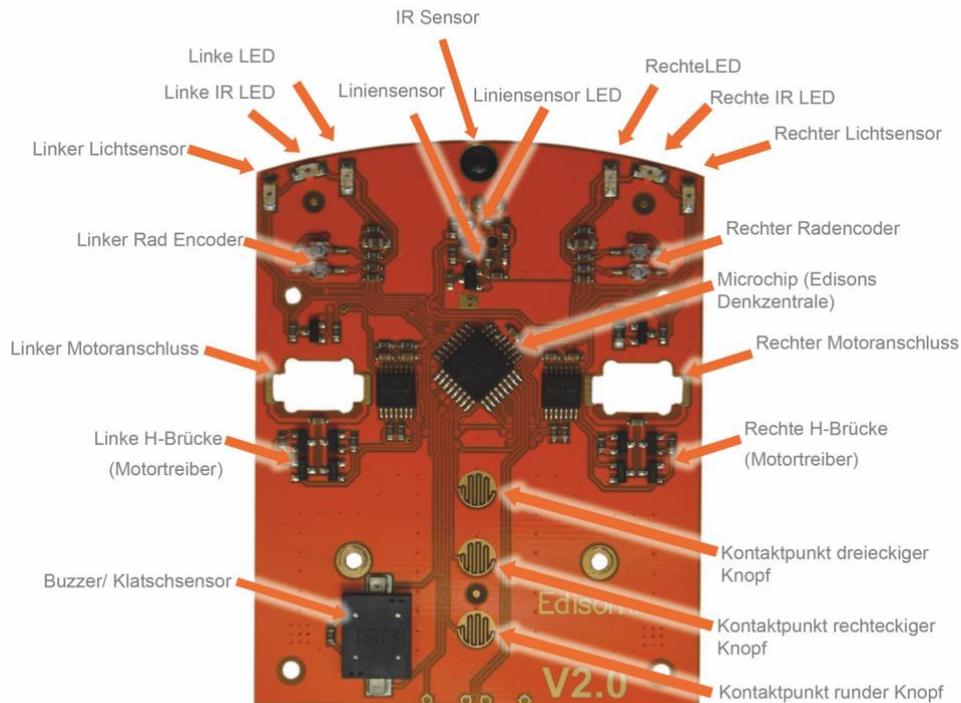
Lektion 1: Edison kennenlernen

Diese erste Lektion dient als Einführung in die Edison-Roboter. Die Entwicklung eines Verständnisses der Teile und Fähigkeiten von Edison wird den Schülern helfen, Edison bei allen Aktivitäten in jeder Einheit effektiv einzusetzen.

In dieser Lektion "treffen" die Schülerinnen und Schüler den Edison-Roboter und lernen die Hauptmerkmale des Roboters kennen, einschließlich des Netzschalters, der Lage und Funktionen der Knöpfe, der Lage und Funktionen der Sensoren sowie der abnehmbaren Teile des Roboters. Die Schülerinnen und Schüler lernen auch die Programmierung der Roboter mit Hilfe von Strichcodes kennen.

Diese Lektion umfasst insgesamt zwei Basisaktivitäten und acht Erweiterungsaktivitäten:

- [U1-1.1 Erforschen wir unsere Edison-Roboter](#)
 - o [U1-1.1a : Ziegel, Blöcke und Edison](#)
 - o [U1-1.2 Lassen Sie uns die Strichcode-Programmierung untersuchen](#)
 - o [U1-1.2a : Sumo-Ringen](#)
 - o [U1-1.2b : Machen Sie Ihren eigenen Strichcode?](#)
 - o [U1-1.2c Ändern Sie ihn nach oben: Strichcodes für TV-Fernbedienung](#)
 - o [U1-1.2d : Edison-Fußball](#)
 - o [U1-1.2e Herausforderung angenommen: Aufbau und Kontrolle des EdTank](#)
 - o [U1-1.2f Herausforderung angenommen: Bau und Steuerung des EdDigger](#)
 - o [U1-1,2g Herausforderung angenommen: Bau und Kontrolle des EdRoboClaw](#)



Aktivität U1-1.1 Erforschen wir unsere Edison-Roboter

Umfang der Tätigkeit	Klein
Empfehlungen zur Bearbeitung	Sehr empfehlenswert, wenn Ihre Schülerinnen und Schüler noch nicht mit Edison-Robotern vertraut sind.
Benötigte Ressourcen	Edison-Roboter mit Batterien, EdComm-Kabel, Arbeitsblatt U1-1.1

Übersicht

In dieser Lektion lernen die Schülerinnen und Schüler den Edison-Roboter kennen, einschließlich der Position der Knöpfe und Sensoren des Roboters. Sie machen sich mit ihrem Roboter und seinen Teilen vertraut, erlangen auf diese Weise ein funktionierendes Verständnis des Roboters und können sich so für den Erfolg in zukünftigen Lektionen vorbereiten.

Tipps und Tricks

- Es kann hilfreich sein, wenn die Schülerinnen und Schüler dieses Arbeitsblatt für ihre Überprüfung der Edison-Komponenten in künftigen Lektionen aufbewahren.

- Auch wenn die Schülerinnen und Schüler in dieser Lektion die EdComm-Kabel nicht mit Programmiergeräten wie Computern oder Tablets verwenden müssen, können Sie ihnen zu diesem Zeitpunkt zeigen, wie sie die Kabel an die Kopfhörerbuchsen der Programmiergeräte anschließen, die sie in zukünftigen Lektionen verwenden werden.
- Bei einigen der Robotikprojekten in späteren Unterrichtsstunden werden die Schülerinnen und Schüler die Räder und das Schleudern von Edison entfernen müssen. Das Verständnis, dass dies möglich ist, kann dazu beitragen, die Fantasie der Lernenden dafür zu wecken, was aus dem Roboter werden könnte. Es wird empfohlen, dass die Schülerinnen und Schüler zumindest das Entfernen der Räder üben. Wenn Sie möchten, dass sie auch das Entfernen der Kufe üben, sollten Sie in Erwägung ziehen, dies über einem Gefäß zu tun, damit sie die Kufe leichter finden können, wenn sie sie fallen lassen.
- Wenn Ihre Schülerinnen und Schüler daran interessiert sind, mehr darüber zu erfahren, was sich auf der Hauptplatine von Edison befindet, können Sie die Layout-Anleitung für die Hauptplatine in voller Größe unter <https://meet Edison.com/content/Edison-Motherboard-layout-V2.pdf> herunterladen.

Lösungsschlüssel

Frage	Typ	Antwortbeispiel	Kennzeichnung von Notizen
1	SE	<i>Als ich Edison einschaltete, leuchteten die roten LEDs auf und begannen zu blinken. Edison machte auch einmal ein zwitscherndes Geräusch.</i>	Bei den Antworten der Schülerinnen und Schüler ist auf zwei wichtige Dinge zu achten: - die roten LED-Leuchten des Roboters leuchten auf und beginnen zu blinken, und - der Roboter macht ein Geräusch.

Aktivität U1-1.1a: Ziegel, Blöcke und Edison

Umfang der Aktivität	Niedrig
Empfehlungen zur Bearbeitung	Wenn die Schülerinnen und Schüler noch nie LEGO-Steine mit Edison verwendet haben, ist dies eine lustige Aktivität, die sie begeistern wird
Benötigte Materialien	Edison-Roboter, EdCreate Bausätze oder jedes andere mit LEGO-Steinen kompatible Bausystem, Arbeitsblatt U1-1.1a

Übersicht

Edison ist so viel mehr als nur ein programmierbarer Roboter im "Auto"-Stil. Die Verwendung von Edison als Bausteinbasis in Robotik-Projekten ist eine der besten Möglichkeiten, die Leistungsfähigkeit von Programmierung und Robotik zum Leben zu erwecken. Diese unterhaltsame MINT- Aktivität ist als risikoarme Aufgabe konzipiert, die den Schülern die Möglichkeit gibt, mit Edison mit „Technik“ zu experimentieren. Die freie Form der Bauaufgabe ermutigt die Schülerinnen und Schüler zu experimentieren, ihre Kreativität zu wecken und damit zu beginnen, einige der taktilen technischen Komponenten des physikalischen Rechnens zu erforschen.

Tipps und Tricks

- Edison funktioniert mit jedem LEGO-Baustein-kompatiblen Bausystem, einschließlich des EdCreate Robot Creator's Kit. Die Steine können oben und unten am Edison-Roboter befestigt werden, und an den Seiten des Roboters können Heringe angebracht werden. Wenn die Räder des Edison-Roboters entfernt werden, können Querachsen in den angetriebenen Steckdosen verwendet werden.
- Vielleicht möchten Sie dieser Aktivität Miniherausforderungen hinzufügen, wie z.B. "Baue so hoch, wie du kannst" oder "Stelle sicher, dass Edisons Knöpfe noch zugänglich sind".
- Sie können die Schülerinnen und Schüler auffordern, in ihrer Antwort anzugeben, warum sie den Entwurf, den sie gemacht haben, erstellt haben.

Lösungsschlüssel

Frage	Typ	Antwort/Beispielantwort	Notiz
1	SE	<i>Mein Edison hatte einen großen Turm auf der Spitze des Roboters.</i>	Dies ist eine Tätigkeit in freier Form.

Aktivität U1-1.2 Untersuchen wir die Strichcode-Programmierung

Umfang der Tätigkeit	Mittel
Empfehlung	Sehr empfehlenswert, wenn Ihre Lernenden neu mit Edison-Robotern arbeiten
Benötigte Materialien	<ul style="list-style-type: none"> - Edison-Roboter mit Batterien, Arbeitsblatt U1-1.2, Aktivitätsblatt U1-1, Taschenlampen, Objekte zur Herstellung von Hindernissen - Fakultativ: EdMats, Tätigkeitsblatt U1-2, Zubehör für die Herstellung eigener Linien/Bordüren

Überblick

Die Schülerinnen und Schüler erforschen einige der Edison-Sensoren in Aktion anhand von voreingestellten Programmen. Zu sehen, wozu die Roboter in der Lage sind, ohne dass eine Programmierung erforderlich ist, ist eine gute Möglichkeit, schon früh Erfolge zu erzielen und die Schülerinnen und Schüler für bevorstehende Aktivitäten zu motivieren.

Diese Aktivität umfasst fünf Programme, die im Speicher von Edison gespeichert sind und auf die mit speziellen Strichcodes zugegriffen wird. Einzelheiten zu jedem durch Strichcodes aktivierten Programm folgen.

Programm 1: Fahren und steuern durch Klatschen

Was geht hier vor? Der Schallsensor von Edison reagiert auf laute Geräusche, wie z.B. Klatschen. Edison dreht sich nach rechts, wenn er ein Klatschen feststellt, oder fährt etwa 30 cm vorwärts, wenn er zwei Klatscher feststellt.

Tipps und Tricks

- Bei hohen Hintergrundgeräuschen kann es für die Roboter schwierig sein, Geräusche zu erkennen und zu unterscheiden. Wenn die Schülerinnen und Schüler mit einem Finger in die Nähe des Geräuschsensors auf der Oberseite ihres Edison-Roboters tippen, hat dies den gleichen Effekt wie Klatschen.

Programm 2: Hindernisse vermeiden

Was geht hier vor? Das Programm zur Vermeidung von Hindernissen nutzt die Infrarot-LEDs (IR-Licht) und den IR-Sensor des Edison-Roboters, um Objekte direkt vor dem Roboter zu erkennen. Sobald das voreingestellte Programm aktiviert ist, fährt der Edison-Roboter vorwärts und dreht sich nach Bedarf, um Hindernissen auszuweichen, auf die er trifft.

Tipps und Tricks

- Hindernisse müssen undurchsichtig, aber nicht zu dunkel (z.B. nicht schwarz) und mindestens so hoch wie Edison sein, damit der Roboter sie erkennen kann.

Programm 3: Einem Licht folgen

Was geht hier vor? In diesem Programm nehmen Edisons zwei Lichtsensoren vorne links und vorne rechts des Roboters Lichtmessungen der Menge an sichtbarem Licht vor, die jeder von ihnen erfasst. Die Messwerte werden dann miteinander verglichen. Wenn die vom rechten Sensor erfasste Lichtmenge höher ist als die des linken Sensors, dann wird Edisons linker Motor vorwärts getrieben und dreht Edison nach rechts, dem Licht entgegen. Diese Bewegung wird fortgesetzt, bis das vom linken Sensor erfasste Lichtniveau den größeren Wert hat. Zu diesem Zeitpunkt stoppt der linke Motor, und der rechte Motor wird vorwärts getrieben und treibt Edison wieder auf das Licht zu.

Tipps und Tricks

- Für die Durchführung dieses Programms benötigen Sie eine Taschenlampe und eine flache Oberfläche, die von anderen hellen Lichtquellen, wie Sonnenlicht oder Leuchtstoffröhren entfernt ist.
- Sobald Edison die helle Lichtquelle "sieht", wird der Roboter auf sie zufahren. Durch Bewegen der Taschenlampe können Sie steuern, wohin Edison fährt.

Programm 4: Einer Linie folgen

Was geht hier vor? Das Linienverfolgungsprogramm verwendet den Auflichtsensor des Edison-Roboters, um Unterschiede zwischen dunklen und hellen Oberflächen unter dem Roboter zu erkennen. Sobald das voreingestellte Programm aktiviert ist, fährt der Edison-Roboter, bis er eine dunkle farbige Linie findet, und folgt dann dieser Linie. In diesem Programm leuchtet der Linienverfolgungssensor des Edison-Roboters von seiner roten LED auf die Oberfläche unter dem Roboter. Der Sensor misst dann die Lichtmenge, die zum Roboter zurückreflektiert wird. Weiße Oberflächen reflektieren viel Licht, was zu einem hohen Lichtpegel führt, während schwarze Oberflächen sehr wenig reflektieren, was zu einem niedrigen Lichtpegel führt. Edison passt die Richtung entsprechend dieser Lichtstärkenmessung an. Wenn Edison nicht auf der Linie ist, dreht er nach rechts, um auf die Linie zu gelangen. Wenn Edison jedoch auf der Linie ist, dreht er nach links, um die Linie zu verlassen. Diese Funktionalität ist der Grund, warum Edison am Rand der Linie hin und her "watschelt".

Tipps und Tricks

- Der Unterschied zwischen den dunklen und hellen Oberflächen muss für den Roboter leicht verständlich sein. Entweder verwenden Sie die dicken, dunklen

Linien auf dem Aktivitätsblatt, die auf einem EdMat (kostenloser Download verfügbar unter <https://meet Edison.com/edmat/>) oder Sie machen eine Spur, der Edison folgen kann, indem Sie eine dunkle (z.B. schwarze) Linie von ca. 1,5 cm Breite auf einem weißen Hintergrund zeichnen.

- Stellen Sie sicher, dass die Schülerinnen und Schüler damit beginnen, Edison neben der schwarzen Linie zu platzieren, aber nicht darüber, so dass der Linienverfolgungssensor auf einer weißen Fläche beginnt. Starten Sie den Roboter immer auf einer weißen Fläche, wenn Sie den Linienverfolgungssensor verwenden.
- Wenn die Räder von Edison den Rand des Papiers des Aktivitätsblatts erwischen, kann dies den Roboter leicht aus der Bahn werfen. Sie können dies beheben, indem Sie das Arbeitsblatt mit Klebeband festkleben oder das Muster des Arbeitsblatts auf einem größeren Stück Papier nachbilden.

Programm 5: Zwischen zwei Linien („Bounce in Borders“)

Was geht hier vor? Wie das Linienverfolgungsprogramm verwendet auch das Programm „Bounce in Borders“ den Auflichtsensor des Edison-Roboters, um Unterschiede zwischen dunklen und hellen Oberflächen unter dem Roboter zu erkennen. Sobald das voreingestellte Programm aktiviert ist, fährt der Edison-Roboter, bis er auf eine dunkle farbige Linie trifft. Dann dreht er sich um und fährt in eine andere Richtung, ohne diese Linie zu überschreiten.

Tipps und Tricks

- Der Unterschied zwischen den dunklen und hellen Oberflächen muss für den Roboter leicht verständlich sein. Entweder verwenden Sie die dicken, dunklen Linien auf dem Aktivitätsblatt, die auf einem EdMat (kostenloser Download verfügbar unter <https://meet Edison.com/edmat/>) oder machen Sie eine Spur, der Edison folgen kann, indem Sie eine dunkle (z.B. schwarze) Linie von ca. 1,5 cm Breite auf einem weißen Hintergrund zeichnen.
- Stellen Sie sicher, dass die Schülerinnen und Schüler damit beginnen, Edison innerhalb der schwarzen Linie zu platzieren, so dass der Linienverfolgungssensor auf einer weißen Fläche steht. Der Roboter kann in der Nähe der schwarzen Linie beginnen, aber nicht darüber. Starten Sie den Roboter immer auf einer weißen Fläche, wenn Sie den Linienverfolgungssensor verwenden.

Aktivität U1-1.2a: Sumo-Ringen

Umfang der Tätigkeit	Klein
Empfehlung	Beenden Sie die Aktivität U1-1.2 vor dieser Aktivität
Benötigte Materialien	- Mindestens 2 Edison-Roboter mit Batterien, Arbeitsblatt U1-1.2a, Tätigkeitsblatt U1-2 - Fakultativ: EdMats, Zubehör für die Herstellung Ihres eigenen Sumo-Rings

Überblick

Bei dieser Aktivität wird einer der speziellen Strichcodes von Edison verwendet, um das Sumo-Ringprogramm zu aktivieren. Das Programm ist bei den Schülern sehr beliebt und eine gute Möglichkeit, eine Atmosphäre der Zusammenarbeit zu schaffen, wenn es um den Einsatz der Roboter geht.

Das Sumo-Ringprogramm kombiniert Aspekte zweier anderer Programme von Edison - das Hüpfen in Grenzen und die Hinderniserkennung. Damit dieses Programm funktioniert, müssen Sie mindestens zwei Edison-Roboter auf einer weißen Fläche mit schwarzem Umriss platzieren. Der Hinderniserkennungsteil des Programms hilft jedem Edison-Roboter, die anderen Roboter zu finden, während der Linienerkennungsteil des Programms Edison dabei hilft, den dunkel gefärbten Rand zu erkennen und darauf zu reagieren.

Tipps und Tricks

- Für diese Aktivität benötigen Sie mindestens zwei Edison-Roboter. Beide Roboter müssen den Barcode des Sumo-Rings scannen, um das Programm zu aktivieren.
- Stellen Sie sicher, dass der Sumo-Ring groß genug ist, damit alle Roboter darin herumfahren können. Wenn der Ring jedoch zu groß ist, dauert es länger, bis die Roboter einander finden.
- Dieses Programm verwendet den Linienverfolgungssensor von Edison. Wenn die Schülerinnen und Schüler mit einem Linienverfolgungsprogramm Schwierigkeiten haben, überprüfen Sie, ob die von ihnen verwendete Linie eine sehr dunkle Farbe, wie z.B. Schwarz, und etwa 1,5 cm (0,6 Zoll) breit ist. Vergewissern Sie sich auch, dass der Hintergrund weiß oder eine andere stark reflektierende Farbe ist. Der EdMat (kostenloser Download verfügbar unter <https://meet Edison.com/edmat/>) funktioniert für diese Aktivität sehr gut.
- Stellen Sie sicher, dass die Schülerinnen und Schüler damit beginnen, Edison innerhalb der schwarzen Linie zu platzieren, so dass der Linienverfolgungssensor auf einer weißen Fläche steht. Der Roboter kann in der Nähe der schwarzen Linie beginnen, aber nicht darüber. Starten Sie den

Roboter immer auf einer weißen Fläche, wenn Sie den Linienverfolgungssensor verwenden.

Aktivität U1-1.2b Einen eigenen Strichcode erstellen

Umfang der Tätigkeit	Niedrig
Empfehlung	- Empfohlen, wenn Lernende neu im Programmieren sind - Beenden Sie die Aktivität U1-1.2 vor dieser Aktivität
Benötigte Materialien	Arbeitsblatt U1-1.2b

Überblick

Diese Offline-Aktivität verdeutlicht, wie die Strichcode-Programmierung mit Edison funktioniert. Die Schülerinnen und Schüler lernen den Zusammenhang zwischen der Funktionsweise von Strichcodes (d.h. durch Aktivierung gespeicherter Programme im Speicher des Roboters) und den Gründen, warum sie keine eigenen Strichcodes erstellen können. Die Schülerinnen und Schüler machen dann eine Übung der Vorstellungskraft - sie erstellen ihr eigenes vorgetäushtes Programm und ihren eigenen Strichcode für Edison.

Tipps und Tricks

- Das Arbeitsblatt beginnt mit einigen Fragen zur Funktionsweise von Strichcodes. Sie können diese Fragen in der Gruppe diskutieren oder die Schülerinnen und Schüler selbst darüber nachdenken lassen, bevor sie die Arbeitsblätter austeilen. Dies ist eine gute Übung zur Extrapolation.
- Lernende ein "gefälschtes" Programm erstellen zu lassen, ist eine gute und risikoarme Aufgabe, um sie dazu zu bringen, darüber nachzudenken, was Programme eigentlich sind. Diese Aktivität bringt die Schülerinnen und Schüler dazu, rechnerisches Denken zu üben, bevor sie explizit in das Konzept eingeführt werden.

Lösungsschlüssel

Frage	Typ	Antwort/Beispielantwort	Hinweis
1	SE	<p>Mein Programm hätte Edison veranlasst, meine saubere Wäsche wegzuräumen. Wenn Edison mein Programm ausgeführt hätte, würde der Roboter zum Wäschekorb fahren und alle sauberen Wäscheteile einsammeln. Dann würde er zu meinem Schrank fahren und alle Kleider wegräumen.</p>	<ul style="list-style-type: none"> - Wenn die Lernenden ein Verständnis für die Fähigkeiten des Roboters entwickelt haben, können sie ein Programm entwerfen, das die tatsächliche Funktionalität des Roboters nutzt. Andernfalls können sie ein Programm entwerfen, das recht phantasievoll ist, wie in der Antwort/Beispielantwort. Beides ist in Ordnung. - Vielleicht möchten Sie sehen, ob die Programme der Schülerinnen und Schüler ein Verständnis des grundlegenden rechnergestützten Denkens, wie z.B. Sequenz, in ihren Programmen zeigen. - Da die Strichcodes nur vorgetäuscht sind, müssen sie nicht wie echte Strichcodes aussehen.

Activity U1-1.2c : TV remote control barcodes

Umfang der Tätigkeit	Niedrig
Empfehlung	- Empfehlen wir dringend, wenn die Schülerinnen und Schüler in späteren Projekten Fernbedienungen mit Edison verwenden werden - Beenden Sie die Aktivität U1-1.2 vor dieser Aktivität
Benötigte Materialien	Edison-Roboter mit Batterien, Arbeitsblatt U1-1.2c, Aktivitätsblatt U1-3, TV- oder DVD-Fernbedienungen)

Überblick

Diese Aktivität führt in die Verwendung von Edison-Robotern mit TV- oder DVD-Fernbedienungen ein. Diese Aktivität ist die erste in der Lektion, in der Edison IR-Messaging verwendet wird, und kann als Gelegenheit dienen, diese Funktionalität zu erforschen.

Die Verwendung von Fernbedienungen mit Edison gibt den Schülerinnen und Schülern die Möglichkeit, ihre Roboter aktiv zu steuern, ohne programmieren zu müssen. Diese Aktivität kann die Schülerinnen und Schüler dazu anregen, mehr darüber zu erfahren, wie sie die Roboter programmieren können.

Tipps und Tricks

- Die Strichcodes der TV/DVD-Fernbedienung wirken sich auf Edison aus, während sich der Roboter im Standby-Modus befindet - stellen Sie sicher, dass die Schülerinnen und Schüler nach der Verwendung der Strichcodes der Fernbedienung NICHT die Abspielen-Taste (Dreieck) drücken. Drücken Sie stattdessen die gekoppelte Taste auf der Fernbedienung, um diese Aktion zu aktivieren.
- Edison ist mit etwa 75% der TV- und DVD-Fernbedienungen kompatibel. Wenn eine Ihrer Fernbedienungen nicht mit Edison funktioniert, versuchen Sie es mit einer anderen Fernbedienung, vorzugsweise einer anderen Marke. Alternativ können Sie eine preiswerte "Universal-Fernbedienung" kaufen und diese auf eine DVD-Fernbedienung von Sony einstellen, die gut mit den Robotern funktioniert.
- Wenn mehrere Schülerinnen und Schüler Programme mit den Fernbedienungscodes in unmittelbarer physischer Nähe zueinander ausführen, kann es zu "Übersprechen" kommen, wenn ein Roboter die von einer anderen Gruppe gesendeten Fernbedienungssignale empfängt und darauf reagiert. Versuchen Sie, die Schülerinnen und Schüler auf Abstand zu halten. Sie können sie auch dazu ermutigen, verschiedene Knöpfe auf ihren Fernbedienungen zu wählen, um sich mit den Robotern zu koppeln und so jede Gruppe von der anderen zu unterscheiden.

Lösungsschlüssel

Frage	Typ	Antwort/Beispielantwort	Hinweis																		
1	SE	<table border="0"> <tr> <td>Programm</td> <td>Fernsteuerbefehl</td> </tr> <tr> <td>Vorwärts fahren</td> <td>Menü-Navigationsrad 'nach oben</td> </tr> <tr> <td>Rückwärts fahren</td> <td>Menü-Navigationsrad 'runter'</td> </tr> <tr> <td>Links drehen</td> <td><i>Kanal nach unten</i></td> </tr> <tr> <td><i>Rechts drehen</i></td> <td><i>Kanal nach oben</i></td> </tr> <tr> <td>nach links drehen</td> <td>Menü-Navigationsrad „links“</td> </tr> <tr> <td>nach rechts drehen</td> <td>Menü-Navigationsrad „rechts“</td> </tr> <tr> <td>Piepton abspielen</td> <td>Lautstärke nach unten</td> </tr> <tr> <td>Melodie abspielen</td> <td>Lautstärke nach oben</td> </tr> </table>	Programm	Fernsteuerbefehl	Vorwärts fahren	Menü-Navigationsrad 'nach oben	Rückwärts fahren	Menü-Navigationsrad 'runter'	Links drehen	<i>Kanal nach unten</i>	<i>Rechts drehen</i>	<i>Kanal nach oben</i>	nach links drehen	Menü-Navigationsrad „links“	nach rechts drehen	Menü-Navigationsrad „rechts“	Piepton abspielen	Lautstärke nach unten	Melodie abspielen	Lautstärke nach oben	<p>Sie sollten überprüfen, ob die Schülerinnen und Schüler die Logik auf ihre Tastenauswahl angewendet haben, um sicherzustellen, dass sie die Funktionsweise des Roboters verstehen.</p>
Programm	Fernsteuerbefehl																				
Vorwärts fahren	Menü-Navigationsrad 'nach oben																				
Rückwärts fahren	Menü-Navigationsrad 'runter'																				
Links drehen	<i>Kanal nach unten</i>																				
<i>Rechts drehen</i>	<i>Kanal nach oben</i>																				
nach links drehen	Menü-Navigationsrad „links“																				
nach rechts drehen	Menü-Navigationsrad „rechts“																				
Piepton abspielen	Lautstärke nach unten																				
Melodie abspielen	Lautstärke nach oben																				

Aktivität U1-1.2d: Edison-Fußball

Umfang der Tätigkeit	Mittel
Empfehlung	- Vollständige Aktivität U1-1.2c vor dieser Aktivität
Benötigte Materialien	<ul style="list-style-type: none"> - Mindestens 2 Edison-Roboter mit Batterien, Arbeitsblatt U1-1.2d, Aktivitätsblatt U1-3, mindestens 2 TV/DVD-Fernbedienungen, Material zur Erstellung des Spielfelds, Tore und Ball - Fakultativ: EdCreate Bausätze/LEGO-Bausteine

Überblick

Bei dieser Aktivität erhalten die Schülerinnen und Schüler eine spezifische Aufgabe, die sie unter Verwendung der Strichcodes der TV-Fernbedienung mit ihren Edison-Robotern lösen müssen.

Die Lernenden müssen zusammenarbeiten, um erfolgreich zu sein. Das Arbeitsblatt weist die Schülerinnen und Schüler ausdrücklich an, gegeneinander anzutreten, und weist dann darauf hin, dass sie zunächst zusammenarbeiten müssen, um ein Übersprechen mit den Fernbedienungen zu vermeiden. Obwohl dies nicht ausdrücklich angegeben ist, müssen die Schülerinnen und Schüler auch zusammenarbeiten, um sich auf die Gestaltung der Spielfläche und der Regeln für

das Fußballspiel zu einigen. Diese Art der natürlichen Zusammenarbeit zu ermutigen und auf sie hinzuweisen, ist eine gute Gelegenheit, Problemlösungen in Aktion zu zeigen.

Die Gestaltung des Spielfelds, der Tore und des Balls sind eine gute Gelegenheit für risikoarme Experimente in einem Szenario der "realen Welt".

Tipps und Tricks

- Die Strichcodes der TV/DVD-Fernbedienung wirken sich auf Edison aus, während sich der Roboter im Standby-Modus befindet - stellen Sie sicher, dass die Schülerinnen und Schüler nach der Verwendung der Strichcodes der Fernbedienung NICHT die Abspielen-Taste (Dreieck) drücken. Drücken Sie stattdessen die gekoppelte Taste auf der Fernbedienung, um diese Aktion zu aktivieren.
- Edison ist mit etwa 75% der TV- und DVD-Fernbedienungen kompatibel. Wenn eine Ihrer Fernbedienungen nicht mit Edison funktioniert, versuchen Sie es mit einer anderen Fernbedienung, vorzugsweise einer anderen Marke. Alternativ können Sie eine preiswerte "Universal-Fernbedienung" kaufen und sie auf eine DVD-Fernbedienung von Sony einstellen, die gut mit den Robotern funktioniert.
- Wenn mehrere Schülerinnen und Schüler Programme mit den Fernbedienungscodes in unmittelbarer physischer Nähe zueinander ausführen, kann es zu "Übersprechen" kommen, wenn ein Roboter ein für einen anderen Roboter bestimmtes Fernbedienungssignal empfängt und darauf reagiert. Erinnern Sie die Schülerinnen und Schüler daran, verschiedene Tasten auf ihren Fernbedienungen zu wählen, um ihre Befehle zu unterscheiden.
- Die Größe des Spielfelds und des Balls beeinflussen den Erfolg der Schülerinnen und Schüler bei dieser Aktivität. Ermutigen Sie die Schülerinnen und Schüler, mit der Einrichtung einer gut funktionierenden Lösung zu experimentieren, bevor ein "offizielles" Spiel stattfindet.
- Eine Alternative zu dieser Aktivität besteht darin, die Schülerinnen und Schüler auf ihren Edison-Robotern mit EdCreate oder alternativ dazu kompatiblen LEGO-Steinen bauen zu lassen, um Vorrichtungen zur Ballkontrolle herzustellen. Stellen Sie sich diese Version der Aktivität eher als modifiziertes Hockey oder Polo denn als Fußball vor!

Aktivität U1-1.2e: EdTank bauen und steuern

Umfang der Tätigkeit	Mittel (einfacher EdTank) Groß (ganzer EdTank)
Empfehlung	- Vollständige Aktivität U1-1.2c vor dieser Aktivität - Die Aktivitäten U1-1.2e, U1-1.2f und U1-1.2g sind zwar alle unterschiedlich aufgebaut, aber das Lernen innerhalb der drei ist vergleichbar. Sie können sich dafür entscheiden, dass die Lernenden nur eine dieser drei Aktivitäten absolvieren.
Benötigte Materialien	1 oder 2 Edison-Roboter mit Batterien pro EdTank (je nach Bauart), 1 TV/DVD-Fernbedienung pro EdTank, Arbeitsblatt U1-1.2e, 1 EdCreate Kit pro EdTank, EdTank-Bauanleitung (verfügbar unter https://meet Edison.com/content/EdCreate/EdBuild-EdTank-instructions.pdf)

Überblick

Die Verwendung von Edison zur Erstellung programmierbarer Roboter-Builds ist eines der aufregendsten Dinge, die Schülerinnen und Schüler mit den Robotern tun können. Die EdBuild-Projekte, bei denen die voreingestellten Anweisungen und EdCreate Bausätze verwendet werden, sind eine ausgezeichnete Möglichkeit, Kinder mit interaktiver Technik vertraut zu machen. Die Verwendung der EdCreate-Bausätze und der Schritt-für-Schritt-Anleitungen zum Erstellen eines EdBuilds erleichtert den Schülern den Einstieg in das Bauen mit Edison, ohne dass sie die Kreation selbst entwerfen müssen.

Über den EdTank:

Der EdTank ist eigentlich zwei Modelle in einem: der Basis-EdTank und der komplette EdTank mit Gummibandkanone.

Der Basis-EdTank verwendet einen Edison-Roboter, der so programmiert werden kann, dass er mit Hilfe von Strichcodes und einer TV- oder DVD-Fernbedienung vorwärts, rückwärts und nach rechts oder links fährt. Der komplette EdTank umfasst einen zweiten Edison-Roboter, der die Gummibandkanone steuert (abfeuert).

Tipps und Tricks

- Die Schülerinnen und Schüler finden es vielleicht am einfachsten, zunächst alle EdCreate-Stücke auf ihrer Arbeitsfläche auszulegen und die Teile in Gruppen desselben Stücktyps und derselben Farbe zu organisieren. Dies

kann den Schülern helfen, die richtigen Teile zu identifizieren und zu verwenden, während sie sich durch den Bau arbeiten.

- Sie müssen die Kanone zurücksetzen und jedes Mal, wenn Sie die Kanone abfeuern, ein Gummiband neu laden. Dies sollte manuell erfolgen, um sicherzustellen, dass das neue Band richtig geladen ist und der Schlagbolzen vollständig in die Ausgangsposition zurückgedrückt wird.
- Die besten Ergebnisse erzielen Sie, wenn Sie die orangefarbenen Bänder, die mit Ihrem EdCreate-Set geliefert werden, als "Munition" in der Kanone verwenden.
- Wie viele echte Panzer ist auch der EdTank so konstruiert, dass er sich nur langsam nach links oder rechts drehen kann. Wenn die EdCreate-Ketten brandneu sind, können sie besonders griffig sein, wodurch der EdTank noch langsamer dreht. Sie können die Griffigkeit verringern, indem Sie die Raupen aus dem EdTank entfernen und sie leicht mit Talkumpuder bestäuben. Achten Sie darauf, überschüssigen Puder von den Raupen abzuklopfen, bevor Sie sie wieder in den EdTank einsetzen.

Lösungsschlüssel

Frage	Typ	Antwort/Beispielantwort	Hinweis
1	SE	<i>Der EdTank drehte sich nicht so gut wie der Edison-Roboter, wenn er sich im "Auto"-Modus befindet. Ich denke, das liegt daran, dass der Panzer einen größeren Wenderadius hat und auch schwerer ist als der normale Edison.</i>	<ul style="list-style-type: none"> - Jede Antwort, die die Logik auf einen Unterschied beim Fahren anwendet, ist akzeptabel. - Es ist möglich, dass sich der EdTank je nach Fahrfläche und Zustand der Raupen so schnell wie der Edison-Roboter dreht.

Aktivität U1-1.2f: Bau und Kontrolle des EdDigger

Umfang der Tätigkeit	Groß
Empfehlung	- Vollständige Aktivität U1-1.2c vor dieser Aktivität - Die Aktivitäten U1-1.2e, U1-1.2f und U1-1.2g sind zwar alle unterschiedlich aufgebaut, aber das Lernen innerhalb der drei ist vergleichbar. Sie können sich dafür entscheiden, dass die Lernenden nur eine dieser drei Aktivitäten absolvieren.
Benötigte Materialien	2 Edison-Roboter mit Batterien pro EdDigger, 1 TV/DVD-Fernbedienung pro EdDigger, Arbeitsblatt U1-1.2f, 1 EdCreate Bausatz pro EdDigger, EdDigger Bauanleitung (verfügbar unter https://meetiedison.com/content/EdCreate/EdBuild-EdDigger-instructions.pdf)

Überblick

Die Verwendung von Edison zur Erstellung programmierbarer Roboter-Builds ist eines der aufregendsten Dinge, die Schülerinnen und Schüler mit den Robotern tun können. Die EdBuild-Projekte, bei denen die voreingestellten Anweisungen und EdCreate Bausätze verwendet werden, sind eine ausgezeichnete Möglichkeit, Kinder mit interaktiver Technik vertraut zu machen. Die Verwendung der EdCreate-Bausätze und der Schritt-für-Schritt-Anleitungen zum Erstellen eines EdBuilds erleichtert den Schülern den Einstieg in das Bauen mit Edison, ohne dass sie die Kreation selbst entwerfen müssen.

Über den EdDigger:

Der EdDigger ist ein ferngesteuerter Bagger mit einer Schaufel, um die man herumfahren kann (vorwärts, rückwärts und rechts oder links drehen). Die Baggerschaufel des EdDigger kann heben und senken und kann kleine Gegenstände, wie z.B. andere Teile aus dem EdCreate Bausatz, tragen.

Tipps und Tricks

- Die Schülerinnen und Schüler finden es vielleicht am einfachsten, zunächst alle EdCreate-Stücke auf ihrer Arbeitsfläche auszulegen und die Teile in Gruppen desselben Stücktyps und derselben Farbe zu organisieren. Dies kann den Schülern helfen, die richtigen Teile zu identifizieren und zu verwenden, während sie sich durch den Bau arbeiten.
- Der obere Roboter verbindet sich mit dem unteren Roboter in der dritten Bolzenreihe von der Vorderseite des unteren Roboters aus. Auf diese Weise überragt der obere Roboter die Rückseite des unteren Roboters um etwa 2 cm.

Lösungsschlüssel

Frage	Typ	Antwort/Beispielantwort	Hinweis
1	SE	<i>Um Objekte aufzusammeln, musste ich den EdDigger die Objekte in etwas schieben lassen, z.B. in eine Wand, so dass die Objekte in den Eimer geschoben werden konnten. Dann konnte ich die Schaufel anheben. Um Objekte aus dem Eimer fallen zu lassen, senkte ich den Eimer zuerst bei gestopptem Edison ab und fuhr dann rückwärts, so dass die Objekte herausfielen.</i>	Jede Antwort, die erklärt, wie die Schaufel betätigt wurde, ist OK.

Aktivität U1-1.2g: Aufbau und Kontrolle des EdRoboClaw

Umfang der Tätigkeit	Groß
Empfehlung	<ul style="list-style-type: none"> - Vollständige Aktivität U1-1.2c vor dieser Aktivität - Die Aktivitäten U1-1.2e, U1-1.2f und U1-1.2g sind zwar alle unterschiedlich aufgebaut, aber das Lernen innerhalb der drei ist vergleichbar. Sie können sich dafür entscheiden, dass die Lernenden nur eine dieser drei Aktivitäten absolvieren.
Benötigte Materialien	2 Edison-Roboter mit Batterien pro EdRoboClaw, 1 TV/DVD-Fernbedienung pro EdRoboClaw, Arbeitsblatt U1-1.2g, 1 EdCreate Bausatz pro EdRoboClaw, Bauanleitung für EdRoboClaw (erhältlich unter https://meet Edison.com/content/EdCreate/EdBuild-EdRoboClaw-instructions.pdf)

Überblick

Die Verwendung von Edison zur Erstellung programmierbarer Roboter-Builds ist eines der aufregendsten Dinge, die Schülerinnen und Schüler mit den Robotern tun können. Die EdBuild-Projekte, bei denen die voreingestellten Anweisungen und

EdCreate Bausätze verwendet werden, sind eine ausgezeichnete Möglichkeit, Kinder mit interaktiver Technik vertraut zu machen. Die Verwendung der EdCreate-Bausätze und der Schritt-für-Schritt-Anleitungen zum Erstellen eines EdBuilds erleichtert den Schülern den Einstieg in das Bauen mit Edison, ohne dass sie die Kreation selbst entwerfen müssen.

Über das EdRoboClaw:

Der EdRoboClaw ist ein ferngesteuerter Gelenkroboterarm, der mit Hilfe der TV/DVD-Strichcodes programmiert und mit einer handelsüblichen TV- oder DVD-Fernbedienung gesteuert werden kann. Sie können den EdRoboClaw vorwärts und rückwärts fahren und ihn nach rechts oder links drehen. Sie können die Klaue auch öffnen und schließen, um einen Gegenstand, z.B. einen der EdCreate-Strahlen, aufzunehmen und zu tragen.

Tipps und Tricks

- Die Schülerinnen und Schüler finden es vielleicht am einfachsten, zunächst alle EdCreate-Stücke auf ihrer Arbeitsfläche auszulegen und die Teile in Gruppen desselben Stücktyps und derselben Farbe zu organisieren. Dies kann den Schülern helfen, die richtigen Teile zu identifizieren und zu verwenden, während sie sich durch den Bau arbeiten.
- Der obere Roboter verbindet sich mit dem unteren Roboter in der zweiten Bolzenreihe von der Vorderseite des unteren Roboters aus. Auf diese Weise übragt der obere Roboter die Rückseite des unteren Roboters um etwa 1 cm.
- Die Klaue besteht aus 3 "Fingern" - zwei parallelen Fingern, die feststehend sind (aus grauen Balken), und dem vordersten Finger, der sich bewegt. Die Zahnradreihe im Gelenkarm steuert diesen vorderen Finger, einschließlich seiner Positionierung relativ zu den stationären Fingern. Die Ausrichtung der beiden vordersten Zahnräder kann sich darauf auswirken, wie der bewegliche Finger relativ zu den stationären Fingern sitzt, wenn die Klaue vollständig geöffnet ist. Wenn die Klaue vollständig geöffnet ist und der EdRoboClaw auf einer flachen Oberfläche (z.B. einem Tisch oder Schreibtisch) sitzt, sollte der bewegliche Finger hoch genug sein, dass einer der grauen EdCreate Balken zwischen Finger und Tisch darunter gleiten kann. Wenn der bewegliche Finger nicht so hoch ist, versuchen Sie, die Vorderseite des Arms vorsichtig zu trennen und das vorderste Zahnrad unabhängig vom nächsten Zahnrad um ein oder zwei Zähne im Uhrzeigersinn zu drehen. Sie werden sehen können, wie sich der vordere Finger dadurch bewegt. Verbinden Sie die Zahnräder und den Arm wieder miteinander.
- Der EdRoboClaw kann Objekte mit einigen flachen Oberflächen aufnehmen. Es kann vorkommen, dass die Schülerinnen und Schüler feststellen, dass sie

runde Gegenstände, wie z.B. einen Stift, nicht so gut aufnehmen und tragen können. Versuchen Sie es mit dem 7-Loch langen grauen Balken aus dem EdCreate Kit.

Lösungsschlüssel

Frage	Typ	Antwort/Beispielantwort	Hinweis
1	SE	<i>Ich konnte meinen Stift nicht mitnehmen, aber ich konnte meinen Müsliriegel und einige Teile von EdCreate mitnehmen. Ich denke, dass die besten Objekte an einigen Stellen flach sind, damit sie nicht aus der Klaue rutschen.</i>	Jede Antwort, die Logik auf die Befunde des Schülers um Objekte herum anwendet, ist akzeptabel.

Lektion 2: EdScratch kennenlernen

Der zweite Teil der ersten Einheit führt die Lernenden in die EdScratch-Programmierungsumgebung ein. Die Schüler lernen, wie man auf die EdScratch Online-Anwendung zugreift, erforschen die Hauptbereiche der Umgebung (einschließlich der 'Bug Box' und der Warnmeldungen) und versuchen, ein Testprogramm von EdScratch auf Edison herunterzuladen.

Diese Lektion hat insgesamt zwei Basisaktivitäten und zwei Erweiterungsaktivitäten:

- [U1-2.1 Lassen Sie uns die EdScratch-Umgebung erkunden](#)
 - o [U1-2.1a Herausforderung angenommen: Laden Sie eine weitere herunter!](#)
 - o [U1-2.1b : Ist EdScratch = Scratch?](#)
- [U1-2.2 Wir erforschen Warnmeldungen](#)

Aktivität U1-2.1 Untersuchen wir die EdScratch-Umgebung

Umfang der Tätigkeit	Groß
Empfehlung	Stark empfohlen, wenn Ihre Lernenden neu bei EdScratch sind
Benötigte Materialien	Basisset, Arbeitsblatt U1-2.1

Überblick

Während die Drag-and-Drop-Programmierung in einer Scratch-ähnlichen Umgebung einigen Lernenden vertraut sein mag, ist das Verständnis des besonderen Layouts und der Funktionalität der EdScratch-Umgebung entscheidend für den Erfolg beim Einsatz der Programmiersprache. Diese Aktivität stellt den Lernenden die Online-Anwendung und das grundlegende Layout von EdScratch vor, einschließlich der Namen der Hauptbereiche innerhalb der Programmierungsumgebung. Die Lernenden üben dann das Herunterladen eines Programms von EdScratch auf ihren Edison-Roboter.

Das Sicherstellen, dass die Schülerinnen und Schüler innerhalb von EdScratch navigieren können und verstehen, wie man EdScratch-Programme auf den Roboter herunterlädt, wird ihre unabhängige Nutzung der Umgebung in zukünftigen Lektionen unterstützen.

Tipps und Tricks

- Einige Geräte, insbesondere Tablets, senken die Lautstärke automatisch ab, wenn sie erkennen, dass ein Audiogerät, z.B. ein Kopfhörer, an die Audiobuchse angeschlossen wurde. Das Programmiergerät kann das EdComm-Kabel als "Kopfhörer" lesen. Vergewissern Sie sich, dass die Lautstärke am Computer oder Tablet nach dem Einstecken des EdComm-Kabels noch ganz aufgedreht ist.
- Während ein Programm heruntergeladen wird, macht Edison ein surrendes Geräusch, ähnlich wie ein altes Wählmodem. Sobald das Programm erfolgreich heruntergeladen wurde, gibt Edison einen zirpenden Piepton ab. Wenn das Programm während des Herunterladens fehlschlägt, gibt Edison ein "Fail-Sound" ab. Wie die Erfolgs- und Misserfolgsgeräusche klingen, können Sie unter <https://meetedison.com/edison-robot-support/trouble-shooting> hören.
- Stellen Sie sicher, dass die Schülerinnen und Schüler den Stecker des EdComm-Kabels erst dann herausziehen, wenn sie das "Erfolgs"-Geräusch hören.
- Wenn viele Schülerinnen und Schüler gleichzeitig Programme herunterladen, kann die Internetgeschwindigkeit langsamer werden, was dazu führt, dass das Programm länger braucht, um die Schaltfläche "Program Edison" im Pop-up-Fenster zu erstellen und das Programm herunterzuladen. Erinnern Sie die Lernenden daran, auf den Erfolgston zu achten, bevor sie das EdComm-Kabel ausstecken, um sicherzustellen, dass sie warten, bis das Programm vollständig heruntergeladen ist.

Lösungsschlüssel

Frage	Typ	Antwort/Beispielantwort	Hinweis
1	EA	<i>In der Menüleiste.</i>	
2	EA	<i>In der Fehlerbox befinden sich keine Meldungen.</i>	Das richtige Test-Programm hat keine Warnmeldungen, so dass in der Fehlerbox nichts erscheinen sollte.

3	SE	<i>Zuerst piepst der Roboter einmal, schaltet die linke LED ein und dreht sich fast ganz nach links. Dann piepst der Roboter einmal, schaltet die rechte LED ein und die linke LED aus und dreht sich wieder in die Startposition zurück. Das Ganze wird fünf Mal wiederholt.</i>	Das Hauptziel besteht darin, dass die Lernenden das richtige Programm herunterladen und erfolgreich ausführen. Wie sie das Programm beschreiben, ist von untergeordneter Bedeutung.
---	----	---	---

Aktivität U1-2.1a: Laden Sie eine weitere herunter!

Umfang der Tätigkeit	Niedrig
Empfehlung	Vollständige Aktivität U1-1.2c vor dieser Aktivität
Benötigte Materialien	- Basisset, Arbeitsblatt U1-2.1a - Abhängig von den ausgewählten Programmen werden zusätzliche Verbrauchsmaterialien benötigt.

Überblick

Diese Aktivität ermutigt die Schülerinnen und Schüler, EdScratch ein wenig weiter unabhängig zu forschen. Durch die Auswahl eines anderen Demoprogramms als das Test-Programm (das in der Basisaktivität verwendet wurde), üben die Lernenden die Schlüsselfertigkeiten, die erforderlich sind, um Programme erfolgreich auf ihre Roboter herunterzuladen.

Die Fragen auf Arbeitsblatt U1-2.1a ermutigen die Schülerinnen und Schüler damit zu beginnen, den Bildschirmcode, den sie im Demoprogramm ihrer Wahl sehen, mit den Aktionen ihres Roboters in der realen Welt zu verknüpfen. Das Verständnis dieser Verbindung ist von grundlegender Bedeutung um in der Lage zu sein, rechnergestützte Denkpraktiken bei der Entwicklung ihrer eigenen Programme effektiv anzuwenden.

Tipps und Tricks

- Sie können die für diese Aktivität erforderliche Versorgung auf das Basisset und das Arbeitsblatt beschränken, indem Sie die Lernenden die Demoprogramme `Moving_with_music` oder `Clap_controlled_driving` auswählen lassen.
- Wenn die Lernenden mit dieser Aktivität Schwierigkeiten haben, lassen Sie sie das Arbeitsblatt U1-2.1 durchsehen.

Lösungsschlüssel

Frage	Typ	Antwort/Beispielantwort	Hinweis
1	SE	<i>Bewegen_mit_Musik</i>	Der Programmname sollte einer aus der Demoliste sein.
2	SE	<i>Ich dachte, der Roboter würde sich zu jeder Musik bewegen, die ich auflege. Stattdessen spielte der Roboter ein Lied und bewegte sich zu diesem Lied.</i>	Alles, was die Frage beantwortet, ist akzeptabel.
3	SE	<i>Der Roboter spielte die Noten, die das Programm enthält. Der Roboter bewegte sich auch auf die gleiche Weise wie das, was die Blöcke sagten. Er führte die Aktionen in der gleichen Reihenfolge aus, in der die Blöcke von oben nach unten stehen.</i>	Idealerweise sollte die Antwort ein Verständnis dafür demonstrieren, dass die Blöcke auf dem Bildschirm mit den Aktionen des Roboters zusammenhängen. Je nach Auswahl des Programms durch die Schülerinnen und Schüler kann dieser Zusammenhang weniger offensichtlich sein.

Aktivität U1-2.1b : Ist EdScratch = Scratch?

Umfang der Tätigkeit	Niedrig
Empfehlung	Empfohlen, wenn Studierende sowohl Scratch als auch EdScratch verwendet haben oder verwenden werden
Benötigte Materialien	Internetverbundene Computer oder Tablets, Arbeitsblatt U1-2.1b

Überblick

Viele Lernenden werden EdScratch sofort 'erkennen' - insbesondere, wenn sie zuvor die MIT-Programmiersprache Scratch verwendet haben. Die EdScratch Anwendung wurde unter Verwendung der vom MIT entwickelten Scratch Blocks Codebasis entwickelt (aufbauend auf der von Google entwickelten Blockly Codebasis). Aus diesem Grund besteht eine so große Ähnlichkeit zwischen Scratch und EdScratch. Es ist jedoch wichtig für die Lernenden zu verstehen, dass es sich bei den beiden Sprachen um getrennte Sprachen handelt. Diese Aktivität bittet die Schülerinnen und Schüler, dies selbst zu erforschen.

Tipps und Tricks

- Diese Aktivität ist möglicherweise nicht so effektiv für Schüler, die Scratch nicht verwendet haben. Sie können sie als optionale Aktivität für jeden Lernenden durchführen, der sich für die Beziehung zwischen den beiden Sprachen interessiert.
- Diese Aktivität eignet sich gut für Gruppenarbeiten oder Arbeiten in der ganzen Klasse, wobei die Lernenden zu einer Liste von Ähnlichkeiten und Unterschieden beitragen.
- Auch wenn Ihre Lernenden Scratch nicht verwenden, kann diese Aktivität eine interessante Erkundung der Tatsache sein, dass es viele verschiedene Programmiersprachen gibt, jede mit ihren eigenen Fähigkeiten und Einschränkungen.

Lösungsschlüssel

Frage	Typ	Antwort/Beispielantwort	Hinweis
1	SE	<p>1 - Die Farben und Formen der Blöcke sehen in beiden Sprachen gleich aus.</p> <p>2 - Sowohl Scratch als auch EdScratch haben eine Blockpalette, eine Menüleiste und einen Bereich zum Erstellen von Programmen.</p> <p>3 - Einige der Blockkategorien, wie "Ton" und "Steuerung", sind in Scratch und EdScratch gleich.</p>	Alle drei Dinge, die in den beiden Sprachen ähnlich oder gleich sind, sind akzeptabel.
1	SE	<p>1 - Scratch verwendet Sprites, wie die Katze, aber EdScratch hat keine Sprites.</p> <p>2 - EdScratch hat eine Bug-Box, aber Scratch hat keine.</p> <p>3 - Die Namen der Blocktypen sind in EdScratch und Scratch unterschiedlich.</p>	Alle drei Dinge, die in den beiden Sprachen unterschiedlich sind oder unterschiedlich erscheinen, sind akzeptabel.

Aktivität U1-2.2 Untersuchen wir die Warnmeldungen

Umfang der Tätigkeit	Niedrig
Empfehlung	<ul style="list-style-type: none"> - Stark empfohlen, wenn Ihre Lernenden neu bei EdScratch sind - Sehr empfehlenswert, wenn Ihre Schülerinnen und Schüler neu mit der Idee des Debuggens in der Programmierung vertraut sind
Benötigte Materialien	Basissatz, Arbeitsblatt U1-2.2

Überblick

Die Fehlerbox in EdScratch liefert Warnmeldungen an die Benutzer. Diese Meldungen können verschiedene Probleme oder potentielle Probleme markieren. In dieser Aktivität lernen die Schülerinnen und Schüler das Fehlerfeld und die zwei Arten von Warnmeldungen kennen, auf die sie im Fehlerfeld stoßen können. Die Idee von Fehlern in der Programmierung wird in dieser Aktivität jedoch nicht explizit vorgestellt, ebenso wenig wie die Praxis des Debuggens.

Das kritische Lernziel dieser Aktivität ist, dass Warnmeldungen in EdScratch keine "Fehler" oder von Natur aus schlechte Dinge sind. Ganz im Gegenteil! Die Fehlerbox und ihre Warnmeldungen sind wunderbare Werkzeuge, um Probleme zu entdecken und zu beheben - kritische Problemlösungs- und Programmierfähigkeiten. Diese Unterrichtsaktivität zielt darauf ab, den Schülerinnen und Schülern zu helfen, eine gute Einstellung gegenüber der Bug-Box zu entwickeln und sie als ein Werkzeug zu betrachten, das sie benutzen können, und gleichzeitig den Rahmen für das Debugging als eine Fertigkeit zu schaffen, die in zukünftigen Unterrichtsstunden entwickelt werden soll.

Tipps und Tricks

- Einige Schülerinnen und Schüler beheben möglicherweise die Fehler, bevor sie die Fragen beantworten. Ein erneutes Laden des Demoprogramms bringt das ursprüngliche Programm und seine Fehler zurück. Mit dem Tastaturkürzel 'Rückgängig' (PC: Strg + z) (Mac: Befehl + z) werden auch Änderungen nacheinander rückgängig gemacht.
- Eine vollständige Anleitung zu den Warnmeldungen finden Sie in EdScratch unter <https://meet.edison.com/robot-programming-software/edscratch/>. Dieser Leitfaden enthält alle Warnmeldungen, was sie bedeuten und Beispiele, wann Sie ihnen begegnen können.

Lösungsschlüssel

Frage	Typ	Antwort/Beispielantwort	Hinweis
1	SE	<i>Es erschien eine Fehlermeldung, die besagt, dass das Programm nicht heruntergeladen werden kann, weil eine rote Warnmeldung angezeigt wird.</i>	Solange eine rote Meldung angezeigt wird, wird das Programm nicht heruntergeladen. Die Antworten der Lernenden müssen zeigen, dass sie dies verstehen.
2	SE	<i>Ich habe den orangefarbenen Block "warte 1 Sekunde" aus den violetten Blöcken herausgenommen.</i>	Das Entfernen des nicht-musikalischen Notenblocks vom Stapel behebt das Problem. Andere Korrekturen, die zum gleichen Ergebnis führen (wie das Löschen aller Blöcke), sind technisch korrekt, aber keine Ideallösung.

3	EA	<i>Die beiden blauen "Fahrblöcke" ("vorwärts um 10 cm bei Geschwindigkeit 2" und "rückwärts um 17 cm bei Geschwindigkeit 5") werden nicht in Edison programmiert.</i>	Diese Antwort basiert auf dem Programm, ohne dass Änderungen vorgenommen wurden.
---	----	---	--

Einheit 2: Los!

Konzentrieren Sie sich in dieser Einheit auf das zentrale rechnerische Konzept der Sequenz. Die Schülerinnen und Schüler erforschen Edisons Fähigkeiten, sich mit Hilfe von Motoren zu bewegen, und nutzen die LEDs des Roboters und den Geräuschsensor/Summer durch eine Reihe von Aktivitäten. Die Grundlagen der Computerprogrammierung einschließlich Ein- und Ausgaben, Bugs und Fehlersuche werden vorgestellt. Die Lernenden beginnen, ihre Vertrautheit mit der Programmierung von Edison in EdScratch und mit der Verwendung von Edison als Basis für kreative Roboter-Builds zu entwickeln.

Lernziele

Die Schülerinnen und Schüler werden:

- in die Idee des rechnergestützten Denkens eingeführt werden und beginnen, bei der Annäherung an Aufgaben rechnergestützte Rahmenwerke zu verwenden
- sich mit dem Begriff der Sequenz vertraut machen
- lernen, was Eingaben, Ausgaben und Eingabeparameter sind
- EdScratch's Hauptausgabeblocke mit Edison zu erforschen, einschließlich Laufwerk, LEDs und Ton
- in der Lage sein, sequentielle Programme für Edison in EdScratch unter Verwendung grundlegender Ausgabeblocke und der Eingabeparameter dieser Blöcke zu erstellen und zu modifizieren
- durch Projekte beginnen, Robotik-Anwendungen in realen Situationen zu erforschen

Kernkompetenzen: rechnergestütztes Denken, Ablauf, Programmierung (Kodierung), Ein- und Ausgaben, Eingabeparameter, Bugs (Fehler), Debugging, Syntax

Lektionen und Aktivitäten in dieser Einheit

Diese Einheit umfasst zwei Lektionen mit insgesamt acht Basisaktivitäten und 13 Erweiterungsaktivitäten.

Lektion 1: Abfolge

- [U2-1.1 Lasst uns erforschen, wie Computer 'denken'](#)
 - o [U2-1.1a : Machen Sie ein PBJ-Sandwich](#)
 - o [U2-1.1b : Menschliche Roboter](#)
- [U2-1.2 Gehen wir in EdScratch Schritt für Schritt vor](#)

- o [U2-1.3 Lasst uns das Fahren mit Edison erkunden](#)
 - [U2-1.3a Herausforderung angenommen: Irrgarten-Wahnsinn](#)
 - [U2-1.3b Herausforderung angenommen: Selbstlaufendes Haustier](#)

Lektion 2: Eingaben und Ausgaben

- [U2-2.1 Lassen Sie uns Edisons Ergebnisse untersuchen](#)
 - o [U2-2.1a Herausforderung angenommen: Sicher durch das Labyrinth fahren](#)
- [U2-2.2 Untersuchen wir die Eingabeparameter](#)
 - o [U2-2.2a : Bringen Sie Edison bei, bis 9 zu zählen](#)
 - o [U2-2.2b Herausforderung angenommen: Bringen Sie Edison bei, laut bis 9 zu zählen](#)
- [U2-2.3 Erforschen wir Edisons musikalische Talente](#)
 - o [U2-2.3a : Ein Lied in einer Runde spielen](#)
 - o [U2-2.3b Herausforderung angenommen: Sie sind der Dirigent](#)
- [U2-2.4 Lasst uns Fehler und deren Behebung untersuchen](#)
- [U2-2.5 Erforschen wir die Motoren von Edison](#)
 - o [U2-2.5a Herausforderung angenommen: Spinnender Garten](#)
 - o [U2-2.5b Herausforderung angenommen: Sich drehendes Sonnensystem](#)
 - o [U2-2.5c Herausforderung angenommen: Kartograph und Navigator](#)
 - o [U2-2.5d : Schriftsteller und Regisseur](#)

Lektion 1: Sequenz

Das Hauptaugenmerk dieser Lektion liegt auf dem grundlegenden rechnergestützten Konzept der Sequenz. Die Schülerinnen und Schüler lernen zunächst, was "rechnergestütztes Denken" ist und warum es ein hilfreicher Rahmen für das Lösen von Problemen und die Arbeit mit Computern ist. Anschließend wird die Sequenz, einer der kritischsten Bausteine des rechnergestützten Denkens, vorgestellt. Die Schülerinnen und Schüler beginnen mit der Anwendung des rechnergestützten Denkens, insbesondere des sequentiellen Denkens, wenn sie an Aufgaben herangehen. Die Aktivitäten in dieser Lektion bieten den Schülern reichlich Gelegenheit, die Anwendung der sequentiellen Logik zu üben und die exakten Schritt-für-Schritt-Anweisungen zu bestimmen, die sie zur Erreichung ihrer Ziele geben müssen. Die Schülerinnen und Schüler beginnen auch, ihre eigenen Programme für Edison zu erstellen, indem sie die "Drive"-Kategorie von Blöcken in EdScratch verwenden.

Diese Lektion hat insgesamt drei Basisaktivitäten und vier Erweiterungsaktivitäten:

- [U2-1.1 Lasst uns erforschen, wie Computer 'denken'.](#)
 - o [U2-1.1a : Machen Sie ein PBJ-Sandwich](#)
 - o [U2-1.1b : Menschliche Roboter](#)
- [U2-1.2 Gehen wir in EdScratch Schritt für Schritt vor](#)
- [U2-1.3 Lasst uns das Fahren mit Edison erkunden](#)
 - o [U2-1.3a Herausforderung angenommen: Irrgarten-Wahnsinn](#)
 - o [U2-1.3b Herausforderung angenommen: Selbstlaufendes Haustier](#)

Aktivität U2-1.1 Untersuchen wir, wie Computer 'denken'.

Umfang der Tätigkeit	Mittel
Empfehlung	Erwägen Sie die Durchführung der Aktivität U2-1.1a vor der Anwendung dieser Aktivität
Benötigte Materialien	Arbeitsblatt U2-1.1, Tätigkeitsblatt U2-1

Überblick

In dieser Offline-Aktivität werden die Lernenden mit den Ideen des rechnergestützten Denkens und der Sequenz vertraut gemacht. Die Lernenden müssen an dieser Stelle nicht vollständig verstehen, was rechnergestütztes Denken beinhaltet, aber es ist wertvoll, den Begriff explizit zu verwenden. Die Verwendung des rechnergestützten Denkens macht das Programmieren viel einfacher, aber es unterscheidet sich ein wenig von der Art und Weise, wie wir normalerweise an Dinge in unserem Leben herangehen und über sie nachdenken. Deshalb ist die Einführung des "rechnergestützten Denkens" als eine Möglichkeit, über die Dinge nachzudenken, ein nützlicher Begriff. Kinder dazu zu bringen, "ihre Computer-Denküte aufzusetzen", kann ihnen helfen, Probleme anders anzugehen und Lösungen zu finden, die sie sonst vielleicht übersehen würden.

Bei dieser Aktivität üben sich die Schülerinnen und Schüler darin, gute aufeinander folgende Anweisungen zu befolgen und zu geben, indem sie jede Handlung explizit ausführen und Schritt für Schritt vorgehen, eine Handlung nach der anderen. Die Entwicklung einer sequentiellen Herangehensweise an die Problemlösung ist entscheidend für gutes rechnergestütztes Denken und Erfolg bei der Programmierung in EdScratch.

Tipps und Tricks

- Die Erweiterungsaktivitäten U2-1.1a und U2-1.1b sind beide darauf ausgerichtet, auf unterhaltsame und einnehmende Weise in die Sequenz einzuführen. Beide Verlängerungsaktivitäten werden am besten als Gruppe oder als ganze Klasse durchgeführt, da diese Aktivitäten im Allgemeinen eine Lehrkraft benötigen, um effektiv durchgeführt werden zu können. Sie können sich dafür entscheiden, eine oder beide Verlängerungsaktivitäten durchzuführen, bevor Sie die Schlüsselkonzepte mit dieser Aktivität formell einführen, oder Sie können diese Aktivitäten nutzen, um die Konzepte nach dieser Aktivität zu vertiefen. Sie könnten z.B. die Aktivität U2-1.1a als eine Klasse durchführen, dann die Schüler einzeln an der Aktivität U2-1.1 arbeiten lassen und dann U2-1.1b als eine Kleingruppen-Verstärkungsaktivität durchführen.

Lösungsschlüssel

Frage	Typ	Antwort/Beispielantwort	Hinweis
1	EA	<i>Das Auto</i>	
2	EA	<i>Das Auto</i>	
3	EA	<i>Die Eiskugel</i>	
4	SE	<i>3 Felder vorwärts bewegen. Nach links drehen. Bewege dich 2 Felder vorwärts.</i>	Jede Antwort, die den Kriterien der Frage entspricht, ist akzeptabel.
5	SE	<i>2 Felder vorwärts bewegen. Nach links drehen. Bewege dich 2 Felder vorwärts. Nach rechts drehen. 1 Feld vorwärts bewegen.</i>	Jede Antwort, die alle Kriterien der Frage erfüllt, ist akzeptabel.
6	SE	<i>Rechts abbiegen. 1 Feld rückwärts gehen. Nach rechts drehen. 1 Feld rückwärts gehen. Nach rechts drehen. 1 Feld rückwärts gehen.</i>	Jede Antwort, die alle Kriterien der Frage erfüllt, ist akzeptabel.

Aktivität U2-1.1a: Ein PBJ-Sandwich machen

Umfang der Tätigkeit	Mittel
Empfehlung	Ziehen Sie in Betracht, diese Aktivität vor der Aktivität U2-1.1 durchzuführen
Benötigte Materialien	Arbeitsblatt U2-1.1a, Zubehör für die Zubereitung des Sandwiches oder alternativer Nahrungsmittel
NB	Diese Aktivität wird am besten als Gruppe oder als ganze Klasse durchgeführt, da sie wahrscheinlich eine Lehrkraft benötigt, um effektiv durchgeführt werden zu können.

Überblick

Diese zweiteilige Offline-Aktivität soll eine lustige Möglichkeit sein, die Dinge schief gehen zu lassen, wenn die Anweisungen nicht genau richtig sind. Entscheidend für den Erfolg bei der Kodierung ist das Verständnis, dass Computer super-wörtlich sind und Anweisungen erfordern, die sowohl anspruchsvoll als auch sequentiell sind. Wenn die Schülerinnen und Schüler ein besseres Verständnis für dieses Konzept entwickeln, können sie bei der Erstellung eigener Programme besser Probleme lösen.

So führen Sie Aufgabe 1 aus

Im ersten Teil dieser Aktivität werden die Schülerinnen und Schüler gebeten, Anweisungen für die Zubereitung eines Sandwichs zu schreiben. Dies kann einzeln oder paarweise erfolgen. Das Arbeitsblatt gibt einige Hinweise darauf, dass diese Aufgabe komplizierter sein könnte, als es zunächst den Anschein hat, aber es ist nicht nötig, auf den Punkt einzugehen. Wenn die Anweisungen nicht ganz so perfekt sind, macht der zweite Teil der Aktivität mehr Spaß und hilft, die Kernaussage der Aktivität an alle Schülerinnen und Schüler zu vermitteln.

So führen Sie Aufgabe 2 aus

Wenn Sie diese Aktivität als ganze Klasse verwenden (empfohlen), sollte diese Aufgabe unter Beobachtung der ganzen Klasse durchgeführt werden. Wählen Sie eine der schriftlichen Anleitungen der Schülerinnen und Schüler entweder nach dem Zufallsprinzip oder suchen Sie sich eine aus, bei der Sie Probleme sehen. Lassen Sie einen Schüler/eine Schülerin jeden Schritt nacheinander von einem Freiwilligen vorlesen. Befolgen Sie jeden Schritt GENAU, wie er geschrieben ist. Wenn in der Anweisung zum Beispiel steht: "Erdnussbutter auf das Brot tun", aber nicht sagt, dass das Glas geöffnet werden soll - öffnen Sie das Glas nicht. Legen Sie einfach das ganze Glas auf das Brot.

Die frühe Auseinandersetzung mit dem, was schief gehen kann, wenn unsere Anweisungen für einen Computer (oder Roboter) in dieser spaßigen Umgebung nicht anspruchsvoll genug sind, hilft Kindern, ein Verständnis für das Konzept zu entwickeln, welches außerhalb einer Programmierung stattfindet. So können Kinder lernen, sequentielles Denken in Betracht zu ziehen und sorgfältig zu analysieren, was vor sich geht, so dass sie Probleme, mit denen sie beim Programmieren konfrontiert werden, effektiver lösen können.

Tipps und Tricks

- Diese Aktivität wird am besten als Gruppe oder ganze Klasse durchgeführt, da sie im Allgemeinen eine Lehrkraft benötigt, um effektiv durchgeführt werden zu können, insbesondere für Aufgabe 2.

- Sie können ein Beispiel für diese Art von Aktivität unter <https://www.youtube.com/watch?v=RjHzD2sfWcQ> sehen.
- Sie können sich dafür entscheiden, diese Aktivität und/oder die Erweiterungsaktivität U2-1.1b auszuführen, um Ihre Untersuchung der Sequenz zu beginnen, bevor Sie das Schlüsselkonzept mit der Aktivität U2-1.1 im Detail untersuchen. Sie könnten z.B. Aktivität U2-1.1a als Klasse durchführen, dann die Schüler einzeln an Aktivität U2-1.1 arbeiten lassen und dann U2-1.1b als Kleingruppen--Verstärkungsaktivität durchführen.
- Wenn Ihre Schülerinnen und Schüler mit Erdnussbutter- und Marmeladenbrot nicht vertraut sind oder wenn Sie Erdnussbutter bei Nussallergien meiden müssen, wechseln Sie das Essen aus. Wählen Sie etwas, das Ihren Schülerinnen und Schülern vertraut ist und das einfach erscheint, aber auch mehrere Schritte erfordert.
 - Alternative Sandwich-Ideen: Aufstriche und Käse, Frischkäse und Gurke, Frischkäse und Marmelade
 - Andere Nahrungsmittel: Reisbällchen, weiche Tacos, Joghurtparfaits

Lösungsschlüssel

Frage	Typ	Antwort/Beispielantwort	Hinweis
1	EA RC SE	<i>Nehmen Sie zwei Stücke Brot. Geben Sie Erdnussbutter auf ein Stück Brot. Das andere Brot mit Marmelade bestreichen. Geben Sie Erdnussbutter und Marmelade zusammen.</i>	Jegliche Anweisungen - auch voller Fehler - sind in Ordnung. Alternativ dazu können Sie die Schülerinnen und Schüler ihre Antworten "fixieren" lassen und diese einreichen, um ein Verständnis der Schlüsselbegriffe "genau" und "Reihenfolge" zu demonstrieren.
2	SE	<i>Es war kein Sandwich - es waren nur Gläser mit Erdnussbutter und Marmelade auf Brot gestapelt!</i>	Meinungsbasierte oder beschreibende Antworten sind akzeptabel.

Umfang der Tätigkeit	Mittel
Empfehlung	Erwägen Sie die Durchführung dieser Aktivität in Verbindung mit U2-1.1
Benötigte Materialien	Arbeitsblatt U2-1.1b, Zubehör für die Markierung des menschlichen Roboterrasters
NB	Diese Aktivität wird am besten als Gruppe oder als ganze Klasse durchgeführt, da sie wahrscheinlich eine Lehrkraft benötigt, um effektiv durchgeführt werden zu können.

Aktivität U2-1.1b: Menschliche Roboter

Überblick

Diese Offline-Aktivität soll den Schülerinnen und Schülern helfen, die 'Roboter'-Logik der Sequenz besser zu verstehen. Ähnlich wie bei der Aktivität U2-1.1 verwenden die Schülerinnen und Schüler bei dieser Aktivität ein Raster, um sequentielle Anweisungen zu geben und zu empfangen. In dieser Aktivität folgen die Schülerinnen und Schüler diesen Anweisungen jedoch als "menschliche Roboter".

Entscheidend für den Erfolg bei der Kodierung ist das Verständnis, dass Computer super-wörtlich sind und sowohl anspruchsvolle als auch sequentielle Anweisungen erfordern. Wenn die Schülerinnen und Schüler ein besseres Verständnis für dieses Konzept entwickeln, können sie ihre eigenen Programme besser lösen.

Was ist zu tun?

Richten Sie ein Gitter in Menschengröße mit einer Zielmarkierung irgendwo auf dem Gitter ein. Bestimmen Sie den Startpunkt und lassen Sie die Schülerinnen und Schüler die erforderlichen Anweisungen ausarbeiten, um den "menschlichen Roboter" zu diesem Ziel zu bringen. Lassen Sie die Schülerinnen und Schüler gemeinsam die Anweisungen aufschreiben und dann einem anderen Schüler, der als Roboter fungiert, die Anweisungen geben. Erinnern Sie die Lernenden daran, dass der Roboter nur die Anweisungen befolgen kann, die der Roboter genau so erhält, wie sie sie erhalten. Kein Fixieren erlaubt!

Tipps und Tricks

- Diese Aktivität lässt sich am besten in Gruppen oder als ganze Klasse durchführen, da sie in der Regel eine Lehrkraft benötigt, um effektiv aufgebaut und durchgeführt werden zu können.
- Ein hervorragendes Beispiel für die Durchführung dieser Aktivität finden Sie unter <https://csunplugged.org/en/topics/kidbots/unit-plan/rescue-mission/>

- Sie können diese Aktivität in Verbindung mit der Aktivität U2-1.1a und der Aktivität U2-1.1 durchführen. Sie könnten z.B. die Aktivität U2-1.1a als eine Klasse ausführen, dann die Schüler einzeln an der Aktivität U2-1.1 arbeiten lassen und dann U2-1.1b als eine Kleingruppen--Verstärkungsaktivität durchführen.

Lösungsschlüssel

Frage	Typ	Antwort/Beispielantwort	Hinweis
1	SE	<i><Schrittfolge, die den Roboter zum Ziel führt></i>	Die Antworten hängen von dem von Ihnen festgelegten Raster und der Aufgabe ab.
2	SE	<i>Ja, aber wir haben vergessen, einige der Wendungen einzubeziehen, also mussten wir das erst einmal korrigieren.</i>	

Aktivität U2-1.2 Gehen wir in EdScratch Schritt für Schritt vor

Umfang der Tätigkeit	Klein
Empfehlung	Empfohlen, wenn Schüler keine Erfahrung mit EdScratch haben.
Benötigte Materialien	Basisset, Arbeitsblatt U2-1.2

Überblick

Diese Aktivität bringt Lernende dazu, die Idee der Sequenz- und Ablaufprogrammierung auf die EdScratch-Umgebung anzuwenden. Die Untersuchung, wie sich eine Änderung der Reihenfolge der Blöcke auf dem Bildschirm auf das Verhalten des Roboters auswirkt, trägt dazu bei, die Bedeutung der Reihenfolge bei der Programmierung zu verfestigen.

Die Idee, dass jeder EdScratch-Block eine 'Aktion' ist, wird vorgestellt. Das Verständnis dieser Idee hilft den Lernenden, die Sequenz beim Erstellen von Programmen anzuwenden. Die Schülerinnen und Schüler experimentieren auch mit dem Bewegen von Blöcken und dem Ändern von Eingaben innerhalb von Blöcken, um Sicherheit bei der Handhabung der EdScratch-Sprache zu gewinnen.

Tipps und Tricks

- Wenn die Schülerinnen und Schüler Schwierigkeiten haben, Blöcke von der Blockpalette zu bewegen, stellen Sie sicher, dass sie die Blöcke direkt nach rechts in den Programmierbereich ziehen und nicht in eine andere Richtung (z.B. nach oben oder unten innerhalb der Blockpalette selbst).
- Blöcke müssen an den Startblock angehängt werden, damit sie heruntergeladen werden können. Schwimmende Blöcke, die nicht mit einem Startblock verbunden sind, werden nicht heruntergeladen.
- Sie können nicht gewünschte Blöcke entfernen, indem Sie sie in den Mülleimer in der unteren rechten Ecke der Programmierbereiche oder zurück in die Blockpalette ziehen.
- Während der Begriff "Eingabeparameter" erst bei der nächsten Aktivität eingeführt wird, ist es eine Gewohnheit, mit den Optionen innerhalb eines Blocks als "Eingabeparameter" zu beginnen, mit der Sie vielleicht sofort beginnen möchten.

Lösungsschlüssel

Frage	Typ	Antwort/Beispielantwort	Hinweis
1	EA	<i>Erstens fährt der Roboter 2,5 Sekunden lang mit der Geschwindigkeit 5 vorwärts, zweitens ertönt ein Piepton. Drittens schaltet er die linke LED ein. Viertens dreht er sich bei Geschwindigkeit 10 um 120 Grad nach links. Fünftens fährt er 2 Sekunden lang mit Geschwindigkeit 2 rückwärts.</i>	ANMERKUNG: Aufgabe 2 ermutigt die Kinder, alle Fehler in dem, was sie entweder in dieser Antwort oder in ihrem Programm geschrieben haben, zu finden und zu beheben.
2	SE	<i>Erstens fährt der Roboter 2,5 Sekunden lang mit Geschwindigkeit 5 vorwärts, zweitens 2 Sekunden lang mit Geschwindigkeit 2 rückwärts und drittens dreht er sich mit Geschwindigkeit 10 um 120 Grad nach links. Viertens: Er schaltet die linke LED ein. Fünftens ertönt ein Piepton.</i>	Die Antworten der Schülerinnen und Schüler sollten nur die vorhandenen Blöcke verschieben, nicht aber Blöcke hinzufügen oder ändern.

Aktivität U2-1.3 Wir erforschen das Fahren mit Edison

Umfang der Tätigkeit	Mittel
Empfehlung	Erwägen Sie, die Aktivität U2-1.3a und diese Aktivität als eine einzige Lektion zu verwenden
Benötigte Materialien	- Baisset, Arbeitsblatt U2-1.3, Tätigkeitsblatt U2-2, Tätigkeitsblatt U2-3 - Optional: Zubehör für die Herstellung eigener Schienen

Überblick

Bringen Sie die Roboter in Bewegung, indem Sie eine sequentielle Programmierung auf die "Laufwerk"-Kategorie der Blöcke in EdScratch anwenden. Die Kinder arbeiten sich durch zwei Programmieraufgaben, um die Roboter dazu zu bringen, Fahraufgaben zu erledigen - zuerst eine gerade Strecke und dann ein Mini-Labyrinth.

Bei der Mini-Labyrinth-Aufgabe müssen die Schülerinnen und Schüler rechnerisches Denken anwenden, um ein sequentielles Programm zur Lösung des Labyrinths zu erstellen. Um das Labyrinth ohne "Schummeln" zu bewältigen, müssen die Schüler unweigerlich verschiedene Versionen ihres ursprünglichen Programmkonzepts iterieren und mit verschiedenen Blöcken und Eingabeparametern experimentieren.

Tipps und Tricks

- Sie können die Schülerinnen und Schüler auch ihre eigenen geraden Spuren oder Labyrinth anfertigen lassen, die sie auf großem Papier oder mit Klebeband auf dem Boden oder Schreibtisch ausfüllen können.
- Erste Versuche in der Programmierung sind fast immer "falsch" - und das ist völlig in Ordnung! Durch Ausdauer und iteratives Verändern von Programmen, durch Tests, während sie laufen, werden die Lernenden zur richtigen Antwort kommen. Belastbarkeit ist der Schlüssel!
- Aufgrund geringfügiger mechanischer Unterschiede bei den Motoren und Encodern in den verschiedenen Edison-Robotern drehen sich manche Roboter nicht genau um 90 Grad, wenn sie die Eingabe von 90 Grad erhalten. Ermutigen Sie die Schülerinnen und Schüler, verschiedene Werte um 90 auszuprobieren (z.B. 87 oder 93), um die Eingabe zu finden, die für ihren Edison am besten funktioniert.

Lösungsschlüssel

Frage	Typ	Antwort/Beispielantwort	Hinweis
1	RC	<p><i>vorwärts für 26,5 cm bei Geschwindigkeit 5</i></p> <p><i>ODER</i></p> <p><i>vorwärts für 10,5 Zoll bei Geschwindigkeit 5</i></p> <p><i>ODER</i></p> <p><i>1,05 Sekunden lang mit Geschwindigkeit vorwärts 5</i></p>	<p>Die Antworten der Schülerinnen und Schüler hängen von den von ihnen gewählten Eingabeparametern ab. Antwort/Beispielantworten gehen davon aus, dass das vorgegebene Aktivitätsblatt mit der Geschwindigkeit 5 ausgefüllt wird.</p>
2	SE	<p><i>1. Gehen Sie geradeaus. 2. Links abbiegen. 3. Gehen Sie geradeaus. 4. Links abbiegen. 5. Gehen Sie geradeaus. 6. Rechts abbiegen. 7. Gehen Sie geradeaus und halten Sie dann an.</i></p>	<p>Es kann sein, dass Lernenden Schritte in ihrem ursprünglichen Plan verpassen - die Erstellung des Programms wird ihnen alle Probleme aufzeigen.</p>

Aktivität U2-1.3a Herausforderung: Irrgarten-Wahnsinn

Umfang der Tätigkeit	Klein bis mittel (abhängig von der Anzahl der versuchten Aufgaben)
Empfehlung	Erwägen Sie, diese Aktivität als eine Reihe von zusätzlichen Herausforderungen für die Aktivität U2-1.3
Benötigte Materialien	<ul style="list-style-type: none"> - Basisset, Arbeitsblatt U2-1.3a, Tätigkeitsblatt U2-3 - Optional: Zubehör für die Herstellung Ihres eigenen Labyrinths

Überblick

Perfekt für Schüler, die die Aktivität U2-1.3 schnell beenden, oder für Schüler, die zusätzliche aufgabenbasierte, sequentielle Programmierübungen benötigen, bietet dieser Satz von 3 Erweiterungsideen Variationen der Labyrinth-Fahraufgabe aus Aktivität U2-1.3.

Tipps und Tricks

- Wenn die Schülerinnen und Schüler ihre eigenen Labyrinth anfertigen, werden sie am erfolgreichsten sein, wenn sie dies auf großem Papier oder mit Klebeband tun, um eine Spur auf dem Boden oder Schreibtisch zu markieren.
- Aufgrund geringfügiger mechanischer Unterschiede bei den Motoren und Encodern in den verschiedenen Edison-Robotern drehen sich manche Roboter bei einer Eingabe von 90 Grad nicht genau um 90 Grad. Ermutigen Sie die Schülerinnen und Schüler, verschiedene Werte um 90 auszuprobieren (z.B. 87 oder 93), um die Eingabe zu finden, die für ihren Edison am besten funktioniert.
- Ermutigen Sie die Schülerinnen und Schüler, sowohl die Eingabeparameter "Drehung" als auch "Schleudern" auszuprobieren, um herauszufinden, welcher am besten funktioniert.

Aktivität U2-1.3b: Laufendes Haustier

Umfang der Tätigkeit	Groß
Empfehlung	Dies ist eine gute Aktivität, um Makerspace/Herstellung und Kodierung zusammenzubringen.
Benötigte Materialien	Basisset, Arbeitsblatt U2-1.3b, Hersteller-Raum-/Flugzeugbedarf

Überblick

Diese lustige Aktivität fordert die Schülerinnen und Schüler auf, ihre Kreativität einzusetzen, um Edison zu etwas anderem zu machen - in diesem Fall zu einem Haustier, das selbst laufen kann.

Der Kodierungsteil dieser Aktivität kann recht einfach sein: ein Programm, bei dem sich der Roboter eine Weile vorwärts bewegt, sich umdreht und wieder zurückkommt, ist ausreichend. Die eigentliche Herausforderung besteht darin, Edison in ein Haustier zu verwandeln.

Ein paar Hinweise, wie dies erreicht werden kann, sind im Arbeitsblatt des Schülers enthalten, aber der Himmel ist die Grenze! Dies ist eine kreative Aktivität für die Schülerinnen und Schüler, um damit zu experimentieren, was in Bezug auf Design, Materialien und Konstruktion funktionieren wird und was nicht.

Tipps und Tricks

- Um das Programm herunterzuladen und auszuführen, müssen die Lernenden auf den EdComm-Kabelanhangsbereich und die Schaltflächen zugreifen. Sie haben die Wahl, die Lernenden zu Beginn der Aktivität darauf hinzuweisen oder alle Entwürfe, die dies nicht vorsehen, als Chance zu nutzen, die Entwurfsplanung zu erkunden.
- Diese Aktivität kann als ein kostengünstiges Heimnäähprojekt angepasst werden.
- Es gibt keine "richtige" Antwort, aber hier sehen Sie, wie eine Musterlösung aussieht:



Lektion 2: Eingaben und Ausgaben

Diese Lektion beginnt mit der Erforschung der grundlegenden Konzepte der Computerprogrammierung von Eingaben (einschließlich Eingabeparametern) und Ausgaben. Die Schülerinnen und Schüler beschäftigen sich mit den verschiedenen Ausgaben der Roboter, indem sie sequentielle Programme für Edison erstellen und modifizieren, indem sie eine Reihe von EdScratch-Blöcken aus den Kategorien "Drive", "LEDs", "Sound" und "Control" verwenden. Die Idee von Fehlern in der Programmierung und die Praxis des Debuggens werden vorgestellt, wodurch die Lernenden darauf vorbereitet werden, bei zukünftigen Aktivitäten mit komplizierteren Programmstrukturen zu arbeiten.

Diese Lektion umfasst insgesamt fünf Basisaktivitäten und neun Erweiterungsaktivitäten:

- [U2-2.1 Lassen Sie uns Edisons Ergebnisse untersuchen](#)
 - o [U2-2.1a Herausforderung angenommen: Sicher durch das Labyrinth fahren](#)
- [U2-2.2 Untersuchen wir die Eingabeparameter](#)
 - o [U2-2.2a : Bringen Sie Edison bei, bis 9 zu zählen](#)
 - o [U2-2.2b Herausforderung angenommen: Bringen Sie Edison bei, laut bis 9 zu zählen](#)
- [U2-2.3 Erforschen wir Edisons musikalische Talente](#)
 - o [U2-2.3a : Ein Lied in einer Runde spielen](#)
 - o [U2-2.3b Herausforderung angenommen: Sie sind der Dirigent](#)
- [U2-2.4 Lasst uns Fehler und deren Behebung untersuchen](#)
- [U2-2.5 Erforschen wir die Motoren von Edison](#)
 - o [U2-2.5a Herausforderung angenommen: Spinnender Garten](#)
 - o [U2-2.5b Herausforderung angenommen: Sich drehendes Sonnensystem](#)
 - o [U2-2.5c Herausforderung angenommen: Kartograph und Navigator](#)
 - o [U2-2.5d : Schriftsteller und Regisseur](#)

Aktivität U2-2.1 Untersuchen wir die Ergebnisse von Edison

Umfang der Tätigkeit	Groß
Empfehlung	
Benötigte Materialien	Basissatz, Arbeitsblatt U2-2.1

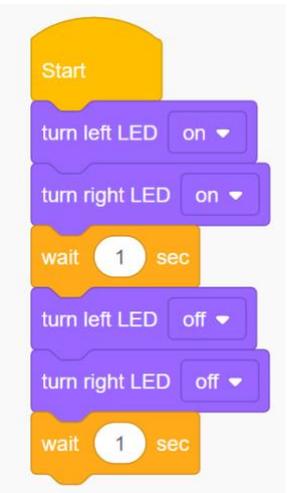
Überblick

In dieser Aktivität wird der Input-Prozess-Ausgabe-Zyklus eingeführt und dann mit ihm experimentiert, wobei der Schwerpunkt auf der Beobachtung der Ausgabeergebnisse der sequentiellen Programmierung liegt. Die Schülerinnen und Schüler beginnen mit der Untersuchung der 'LED'- und 'Sound'-Kategorien von Blöcken in EdScratch und werden in ihre erste Kontrollstruktur - den 'Warte'-Block - eingeführt.

Der Beginn des Verständnisses, dass der Input-Prozess-Output-Zyklus in jedem Programm im Spiel ist, hilft den Lernenden, die Programmierung in einzelne Phasen zu zerlegen. Dies ist eine wichtige Fähigkeit für das Debugging in der Programmierung und für die rechnerische Denkfähigkeit, Probleme in kleinere Blöcke zu zerlegen.

Lösungsschlüssel

Frage	Typ	Antwort/Beispielantwort	Hinweis
1	EA	<input type="checkbox"/> <i>Edisons Lichter ausschalten - - LEDs</i> <input type="checkbox"/> <i>Spielen Sie eine Musiknote -- Ton</i> <input type="checkbox"/> <i>Spin Edison rechts -- Antrieb</i>	
2	SE	<i>Der Roboter führte alle Schritte aus, aber er war wirklich schnell, so dass es so aussah, als ob er sie nicht der Reihe nach, sondern auf einmal ausführte.</i>	Die Lernenden können feststellen, dass die Ausführung der Ergebnisse sehr schnell erfolgt.

3	RC		<p>Was "am besten" funktioniert, ist subjektiv, aber die Lernenden sollten mindestens eine Wartezeit zwischen dem An- und Ausschalten der LEDs haben.</p>
Mini-challenge			<p>Das Arbeitsblatt weist die Schülerinnen und Schüler nicht an, ihre Antwort mitzuteilen, aber Sie können sich dafür entscheiden, dass die Schülerinnen und Schüler ihre Gedanken oder den daraus resultierenden Code präsentieren.</p>

Aktivität U2-2.1a Herausforderung annehmen: Das Labyrinth sicher befahren

Umfang der Tätigkeit	Klein
Empfehlung	Schließen Sie die Aktivitäten U2-1.3 und U2-2.1 vor dieser Aktivität ab
Benötigte Materialien	Basisset, Arbeitsblatt U2-2.1a, Tätigkeitsblatt U2-3

Überblick

Die Lernenden müssen ihr Verständnis von Sequenzen bei dieser Aktivität sorgfältig in die Praxis umsetzen. Das Schreiben eines Programms, bei dem der Roboter alle Schritte zur Vervollständigung des Labyrinths ausführt und die Drehrichtung mit Hilfe der LED-Leuchten, des Summers oder beidem anzeigt, bevor er jede Richtungsänderung vornimmt, ist eine großartige Möglichkeit, die Schüler ihr Verständnis der Sequenz in einer kreativeren Programmieraufgabe anwenden zu lassen.

Tipps und Tricks

- Erinnern Sie die Schülerinnen und Schüler daran, dass sie "Warteböcke" verwenden können, um den Roboter für eine bestimmte Zeitspanne pausieren zu lassen.
- Während das Arbeitsblatt die Schülerinnen und Schüler nicht auffordert, ihr Programm zu erfassen, können Sie den Schülerinnen und Schülern die Möglichkeit geben, ihr Programm aufzuschreiben oder vorzuführen und ihre Gründe für die Verwendung der gewählten Böcke, Eingaben und Sequenz zu erläutern.
- Aufgrund geringfügiger mechanischer Unterschiede bei den Motoren und Encodern in den verschiedenen Edison-Robotern drehen sich einige Roboter nicht genau um 90 Grad, wenn sie die Eingabe von 90 Grad erhalten. Ermutigen Sie die Schülerinnen und Schüler, verschiedene Werte um 90 (z.B. 87 oder 93) auszuprobieren, um die Eingabe zu finden, die für ihren Edison am besten funktioniert.
- Ermutigen Sie die Schülerinnen und Schüler, sowohl die Eingabeparameter "Drehung" als auch "Schleudern" auszuprobieren, um herauszufinden, welcher am besten funktioniert.

Activity U2-2.2 Let's explore input parameters

Umfang der Tätigkeit	Klein
Empfehlung	
Benötigte Materialien	- Arbeitsblatt U2-2.2 - Fakultativ: Set Grundausstattung

Überblick

Diese Offline-Aktivität untersucht das Konzept der Eingabeparameter und baut auf der in Aktivität U2-1.3 eingeführten Basisdefinition auf. Diese Aktivität hilft den Schülerinnen und Schülern, das Konzept der Eingabeparameter zu verstehen, und schafft einen Rahmen, um herauszufinden, was ein bestimmter Eingabeparameter zu einem Programm beiträgt. Die Fähigkeit, kritisch über Eingabeparameter nachzudenken, hilft den Lernenden zu verstehen, was in einem Programm tatsächlich geschieht.

Eingabeparameter tragen kritische Informationen zu den Eingaben von Programmen bei. Durch die explizite Untersuchung von Eingabeparametern gewinnen die Lernenden ein tieferes Verständnis für die Eingaben, die sie in ihren Programmen verwenden. Die Fähigkeit, die Eingabekomponenten eines bestimmten Programms mit den erwarteten Ergebnissen dieses Programms zu verknüpfen, ist eine wichtige Fähigkeit, die für das computergestützte Denken und Problemlösen zu entwickeln ist.

Tipps und Tricks

- Möglicherweise möchten Sie, dass die Schülerinnen und Schüler die Programme im Arbeitsblatt mit ihren Edison-Robotern verwenden, um die Programme in Aktion zu sehen, was dazu beiträgt, die Verbindung zwischen Ein- und Ausgaben zu festigen.
- Um den Zusammenhang zwischen den Auswirkungen der Eingabeparameter auf die Ausgaben weiter zu demonstrieren, lassen Sie die Schülerinnen und Schüler die Eingabebedingungen der Arbeitsblattprogramme in EdScratch anpassen, stellen Sie Hypothesen darüber auf, was diese Änderungen an den Eingaben auf die Ausgaben wirken, und testen Sie dann die Programme mit ihren Robotern, um die Ergebnisse zu sehen.

Lösungsschlüssel

Frage	Typ	Answer	Hinweis
1	EA	<i>Sekunden</i>	
2	EA	7	
3	EA	<i>Fahren Sie 6,5 Sekunden lang mit Geschwindigkeit 7 rückwärts.</i>	
4	EA	4	Dieses Programm hat zwar mehrere Blöcke, aber nur einer der Blöcke hat Eingabeparameter.
5	EA	<i>In welche Richtung soll der Roboter gehen?</i>	Jede Antwort, die ein Verständnis dafür zeigt, dass dieser Eingabeparameter mit der Richtung zusammenhängt, ist akzeptabel.

Aktivität U2-2.2a : Edison lehren, bis 9 zu zählen

Umfang der Tätigkeit	Mittel
Empfehlung	Vollständige Aktivität U2-2.2 vor dieser Aktivität
Benötigte Materialien	Basisset, Arbeitsblatt U2-2.2a, Tätigkeitsblätter U2-4, U2-5, U2-6 und U2-7

Überblick

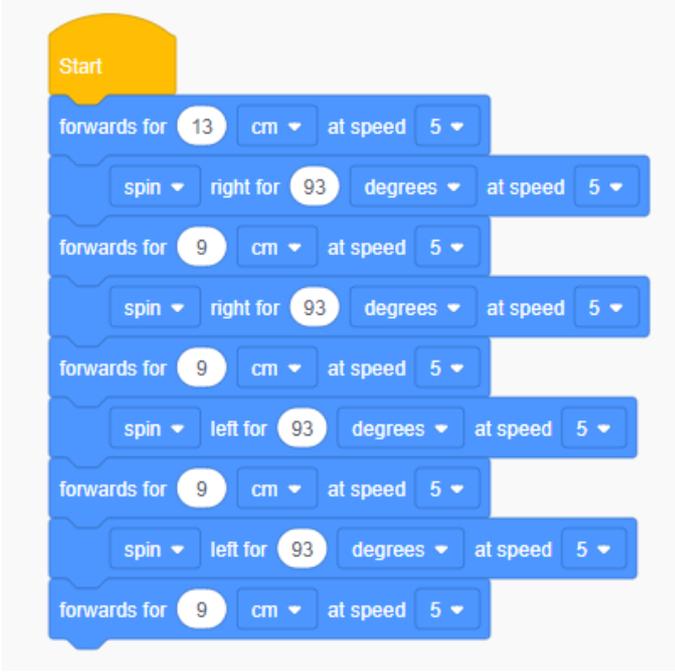
Diese Erweiterungsaktivität bietet Programmierherausforderungen, bei denen einige der Fähigkeiten, die die Lernenden entwickelt haben, zum Einsatz kommen. Das Erstellen von Programmen für den Edison-Roboter, um die Nummern der Digitalanzeige ihrer Wahl zu "verfolgen" (d.h. im Muster zu fahren), erfordert die

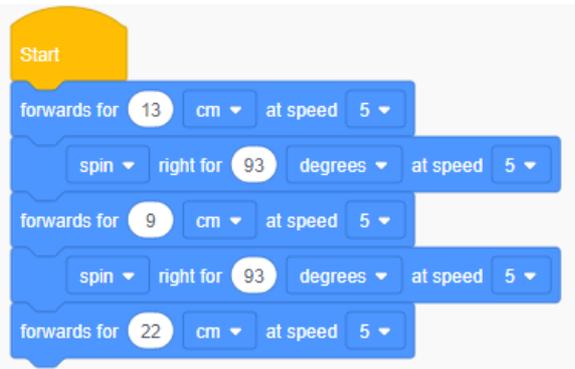
Anwendung von sequentiellem Denken und ein Verständnis des Input-Prozess-Output-Zyklus.

Diese Aktivität lenkt die Aufmerksamkeit auf einen bestimmten Eingabeparameter (Abstand in den Antriebsblöcken) und fordert die Schülerinnen und Schüler auf, Muster hinsichtlich der Auswirkungen des Eingabeparameters auf die Ausgabe zu identifizieren. Die Schülerinnen und Schüler extrapolieren dann aus dem, was sie beobachten, und wenden ihr Verständnis auf eine andere Programmaufgabe an.

Tipps und Tricks

- Jedes Segment in allen Aktivitätsblättern der Digitalanzeige hat die gleiche Länge. Wenn sich Edison jedoch dreht, dreht sich der Roboter nicht auf der Stelle, sondern fährt ein Stück weiter. Infolgedessen ist der Abstandsparameter, der erforderlich ist, um ein Segment nach einer Drehung abzuschließen, niedriger als der Abstandsparameter, der erforderlich ist, um ein gerades Segment abzuschließen.
- Aufgrund geringfügiger mechanischer Unterschiede in den Motoren und Encodern innerhalb der verschiedenen Edison-Roboter drehen sich einige Roboter bei der Eingabe von 90 Grad möglicherweise nicht genau um 90 Grad. Ermutigen Sie die Schülerinnen und Schüler, verschiedene Werte um 90 herum auszuprobieren (z.B. 87 oder 93), um die Eingabe zu finden, die für ihren Edison am besten funktioniert.
- Ermutigen Sie die Schülerinnen und Schüler, sowohl die Eingabeparameter "Drehung" als auch "Schleudern" auszuprobieren, um herauszufinden, welcher am besten funktioniert.
- Es ist möglich, ein Programm für jede der Zahlen in den Aktivitätsblättern zu schreiben, ohne dass der Roboter dasselbe Segment zweimal durchlaufen muss. Sie können dies erwähnen, um den Lernenden zu helfen, sich die Sequenz vorzustellen, die sie in ihren Programmen verwenden müssen.
- Wenn die Räder von Edison den Rand des Papiers des Arbeitsblatts erwischen, kann dies den Roboter leicht aus der Bahn werfen. Sie können dies beheben, indem Sie das Arbeitsblatt mit Klebeband festkleben oder das Muster des Arbeitsblatts auf einem größeren Blatt Papier nachbilden.

Frage	Typ	Antwort/Beispielantwort	Hinweis
1	SE	2	
2	RC		
3	SE	<p><i>Der Abstandeingabeparameter für den ersten Antriebsblock ist etwas länger als alle anderen. Alle anderen Sätze verwenden die gleiche Nummer für den Abstandeingabeparameter. Ich denke, dies bedeutet, dass jede Nummer der Digitalanzeige nur einen der beiden Abstandeingabeparameter benötigt: einen für ein gerades Segment und einen für nach einer Kurve.</i></p>	<p>Im Idealfall wird in den Antworten der Lernenden eine Korrelation zwischen dem in einem Eingabeparameter benötigten Wert und der wiederholten realen Distanz, die der Roboter zurücklegen muss, festgestellt.</p>
4	SE	7	

5	RC		<p>Sie können wählen, ob die Lernenden in jedem Programm dieselben Entfernungseingabeparameter verwenden.</p>
6	SE	<p><i>Der Abstandeingabeparameter für den ersten Antriebssatz, der gerade ist, ist in beiden Programmen gleich. Alle auf eine Kurve folgenden Fahrsätze sind kürzer, aber untereinander gleich. Der Abstandeingabeparameter in den Kurven ist immer derselbe. Der letzte Satz des Programms für '7' ist in jedem Programm der gleiche wie der erste Antriebssatz in jedem Programm plus der Antriebssatz nach einer Kurve, da er eine Kombination aus zwei Segmenten ist.</i></p>	<p>Im Idealfall wird in den Antworten der Lernenden eine Korrelation zwischen dem in einem Eingabeparameter benötigten Wert und der wiederholten realen Distanz, die der Roboter zurücklegen muss, festgestellt.</p>

Lösungsschlüssel

Aktivität U2-2.2b : Bringen Sie Edison bei, laut bis 9 zu zählen

Umfang der Tätigkeit	Klein
Empfehlung	Es wird empfohlen, die Aktivität U2-2.2a vor dieser Aktivität abzuschließen.
Benötigte Materialien	- Baisset, Arbeitsblatt U2-2.2b, Tätigkeitsblätter U2-4, U2-5, U2-6 und U2-7 - Optional: Verbrauchsmaterial zur Herstellung Ihrer eigenen Digitalanzeigenummern einschließlich Anhang 1 (leeres Digitalanzeigenblatt) aus diesem Leitfaden

Überblick

Bringen Sie bei dieser halboffenen Herausforderung mehrere Ausgaben in einem einzigen sequentiellen Programm zusammen.

Es gibt keinen richtigen" oder falschen" Weg, diese Programmieraufgabe abzuschließen. Jedes Programm, das die Kriterien 'Fahren Sie die Zahl' und 'Zählen Sie die Zahl' erfüllt, ist in Ordnung. Dies ist ein wichtiges Konzept, das die Lernenden zu verstehen beginnen. Programmieren ist sehr kreativ, und es sind oft viele Lösungen möglich. Das ist ein Teil dessen, was Informatik so lustig und interessant macht!

Tipps und Tricks

- Die Schülerinnen und Schüler können die Soundausgänge, die LED-Ausgänge, beide oder eine Kombination von beiden zum Zählen verwenden. (Sie können wahrscheinlich auch die Drive-Ausgänge zum Zählen verwenden, wenn sie einen Weg finden, dies zu tun, während sie noch ihre Anzeigenummer zurückverfolgen können). Wenn der Student zum Beispiel die Zahl 5 gewählt hat, könnte sein Edison ein Segment der Anzeigenummer ansteuern, piepsen, drehen, dann das nächste Segment ansteuern, piepsen, drehen usw., bis er alle Segmente angesteuert und fünfmal gepiept hat. Alternativ könnte das Programm den gesamten Pfad fahren und dann die linke LED des Roboters fünfmal ein- und ausschalten. Jedes Programm, bei dem der Roboter die Bahn fährt und irgendwie den gleichen Wert wie diese Bahn signalisiert, ist in Ordnung!
- Wenn Sie die Idee hervorheben möchten, dass viele verschiedene Lösungen möglich sind, teilen Sie Ihre Schülerinnen und Schüler für diese Aktivität in Gruppen ein und geben Sie jeder Gruppe die gleiche Nummer der Digitalanzeige. Lassen Sie jede Gruppe ihre Lösung für die Klasse demonstrieren. Diskutieren Sie die Unterschiede und Ähnlichkeiten im Programm jeder Gruppe und zeigen Sie auf, wie sie, auch wenn sie

unterschiedlich sind, durch die Erfüllung der Kriterien dennoch alle richtig sind.

- Wenn Sie die Schülerinnen und Schüler ihre eigenen Zahlen machen lassen, können Sie ihnen einen Hinweis geben, mit welchen Segmenten sie arbeiten müssen: Ein Standard-Anzeigegitter ist voll beleuchtet, wenn die Zahl 8 angezeigt wird. Alternativ können Sie ihnen auch das leere Arbeitsblatt mit den Nummern der Digitalanzeige zur Verfügung stellen, das in Anhang 1 dieses Leitfadens zu finden ist.

Activity U2-2.3 Erforschen wir Edisons musikalische Talente

Umfang der Tätigkeit	Groß
Empfehlung	
Benötigte Materialien	Basissatz, Arbeitsblatt U2-2.3

Überblick

Einfache musikalische Melodien sind ein großartiges Beispiel für eine Sequenz in Aktion. Die Verwendung von Edison zur Erstellung von Musik verbindet Sequenz und Input-Output in Programmen, die die Schülerinnen und Schüler lieben.

Diese Aktivität ist darauf ausgerichtet, das Vertraute mit dem Unbekannten zu vermischen. Indem die Schülerinnen und Schüler ohne ausdrückliche Anweisungen oder detaillierte Erklärungen mit neuen Blöcken experimentieren, wenden sie rechnergestützte Denkmethoden an und entwickeln eine Denkweise zur Problemlösung.

Die letzte Aufgabe in dieser Aktivität beinhaltet eine halboffene Programmieraufgabe. Dies ist eine großartige Gelegenheit für die Lernenden, die kreative Seite des Programmierens zu erleben.

Tipps und Tricks

- Das in Aufgabe 1 des Arbeitsblatts verwendete Lied ist Mary Had a Little Lamb. Verwenden Sie diesen Freigabecode, um auf ein Programm mit mehr von dem Lied zuzugreifen: <https://www.edscratchapp.com?share=Eb12x3Dm>
- Der dritte Eingabeparameter im Block 'Musiknote' hat drei Auswahlmöglichkeiten: (-), scharf, flach. Die Standardeingabe (-) bedeutet, dass eine normale Note gespielt werden soll, nicht eine scharfe oder flache.

- Der "Tempo"-Block legt das zu spielende Tempo fest. Nur Blöcke, die nach einem Tempoblock kommen, sind betroffen.
- Das Standardtempo von Edison ist "mittel". Wenn kein Block "set music tempo to" verwendet wird, spielt der Roboter die Noten mit dem Standardtempo.
- Verwenden Sie diesen Freigabecode, um auf ein Programm mit mehr von The Hokey-Pokey song zuzugreifen:
<https://www.edscratchapp.com?share=v0wMx5D5>

ANMERKUNG: Diese Version des Liedes verwendet Schleifen.

- Sie sollten mit der Verwendung dieses Programms warten, bis die Lernenden die Wiederholungsschleifen studiert haben.
- Wiederholungsschleifen können nicht innerhalb von 'Abspielen Music in Background'-Blöcken verwendet werden. Um die längere Version des Liedes im Hintergrund zu verwenden, duplizieren Sie die Blöcke innerhalb der Schleife und fügen Sie sie stattdessen in sequentieller Reihenfolge in das Programm ein.

Lösungsschlüssel

Frage	Typ	Antwort/Beispielantwort	Hinweis
1	SE	<i>Sehr schnell</i>	
2	SE	<i>Dadurch wurde das Lied im Vergleich zum Original mit dem Eingabeparameter 'mittel' viel schneller abgespielt.</i>	
3	SE	<i>Der Block 'Tempo festlegen' legt die Geschwindigkeit oder das Tempo der Musik fest.</i>	Die Lernenden sollen erkennen, dass der Block das Tempo der Musik bestimmt.

4	EA	<p><i>Wenn der Block 'Tempo festlegen' am Ende steht, wird er nichts bewirken. Das liegt daran, dass Edison sich die Blöcke einzeln ansieht und sie in der Reihenfolge durchführt. Wenn es keine Musikblöcke nach dem Tempoblock gibt, dann hat der Tempoblock keine Wirkung.</i></p>	<p>Im Idealfall erkennen die Schülerinnen und Schüler, dass sequenzielle Programmierung bedeutet, dass die Dinge nur der Reihe nach in Kraft treten. Ziehen Sie in Erwägung, dass die Lernenden versuchen, die Tempoblöcke an verschiedenen Stellen im Programm zu verwenden, um dieses Konzept zu testen und zu verstärken.</p>
5	EA	<p><i>Nein, es spielte die Musik und bewegte sich dann. Ich glaube, das liegt daran, dass erst die Blöcke mit der ganzen Musik und dann die Laufwerksblöcke in Ordnung sind.</i></p>	<p>Die Schülerinnen und Schüler sollten bei ihren Antworten beachten, dass sich der Roboter erst nach dem Abspielen der Musik bewegt, und die Reihenfolge als Grund dafür angeben.</p>

Aktivität U2-2.3a : Ein Lied in einer Runde spielen

Umfang der Tätigkeit	Mittel
Empfehlung	Vollständige Aktivität U2-2.3 vor dieser Aktivität
Benötigte Materialien	Grundausstattungs-Set, Arbeitsblatt U2-2.3a, Notenblätter oder Zugang für Schüler zum Nachschlagen von Liedern
NB	Diese Aktivität wird am besten als Gruppe oder ganze Klasse durchgeführt.

Überblick

Die Zusammenarbeit mit anderen Lernenden an einem Programmierprojekt steht im Mittelpunkt dieser Aktivität. Durch Hervorhebung der "Ton"-Ausgaben ihrer Roboter und Verwendung der "Warte"-Blocksteuerungsstruktur arbeiten die Schülerinnen und Schüler zusammen, um mehrere Roboter zu einer harmonischen Leistung zu bringen.

Tipps und Tricks

- Erwägen Sie, Ihren Schülerinnen und Schülern einige Lieder oder Notenblätter für diese Aktivität zur Verfügung zu stellen. Gute Optionen für Lieder in einer Runde umfassen:
 - *Row, Row, Row Your Boat*
 - *Three Blind Mice*
 - *Frère Jacques*
 - *Farmer in the Dell*

- Es kann schwierig sein, das richtige Timing zu finden, damit die Roboter perfekt harmonisieren, da alle Roboter genau zur gleichen Zeit starten müssen. Aktivität U5-1.4d : Der Edison-Chor greift dieses Problem auf, indem er die "Warte"-Blöcke für IR-Nachrichtenblöcke ausschaltet. Sie können sich dafür entscheiden, mit der Aktivität U5-1.4d zu warten: Der Edison-Chor und diese Aktivität zusammen, wobei die Vor- und Nachteile jedes Ansatzes hervorgehoben werden.

- Wenn Sie wirklich musikalisch werden wollen, bringen Sie Ihre Schülerinnen und Schüler dazu, mit ihren Edison-Robotern mitzusingen, so dass die Schülerinnen und Schüler und die Roboter das Lied in einer Runde gemeinsam spielen!

Aktivität U2-2.3b : Sie sind der Dirigent

Umfang der Tätigkeit	Mittel
Empfehlung	Vollständige Aktivität U2-2.3 vor dieser Aktivität
Benötigte Materialien	Grundausstattungs-Set, Arbeitsblatt U2-2.3b, Notenblätter oder Zugang für Schüler zum Nachschlagen von Liedern

Überblick

Diese kostenfreie Aktivität fordert die Schülerinnen und Schüler auf, ihre Lieblingsmelodie zum Leben zu erwecken, indem sie sie in EdScratch programmieren, damit Edison sie spielen kann. Musik ist eine großartige Möglichkeit für Kinder, sich selbst und ihre Interessen auszudrücken, und diese Aktivität ermöglicht es ihnen, diesen Enthusiasmus mit dem Programmieren zu verbinden und dabei zu helfen, die Künste in MINT zu präsentieren.

Tipps und Tricks

- Wenn Schülerinnen und Schüler online nach ihrer Lieblingsmusik suchen, eröffnet sich die Möglichkeit, über sichere Suchgewohnheiten, Urheberrecht und Zuschreibung und viele andere Themen der digitalen Verantwortung zu diskutieren.
- Schülerinnen und Schüler mit musikalischem Talent können sich dafür entscheiden, ihren eigenen Originalsong zu programmieren, anstatt einen bereits vorhandenen Song zu verwenden.

Aktivität U2-2.4 Untersuchen wir Fehler und deren Behebung

Umfang der Tätigkeit	Mittel
Empfehlung	Wird dringend empfohlen, bevor die Lernenden beginnen, Kontrollstrukturen zu programmieren (einschließlich Schleifen und Bedingungen) oder in Programmen zu erfassen.
Benötigte Materialien	Basisset, Arbeitsblatt U2-2.4

Überblick

Machen Sie sich darauf gefasst - diese Lektion hat viel mehr Text und viel mehr Definitionen als die meisten anderen, aber lassen Sie sich davon nicht abschrecken! Fehler zu verstehen und zu debuggen ist eine der wichtigsten Fähigkeiten, die Sie Ihren Schülern vermitteln müssen. Beim computergestützten Denken geht es in vielerlei Hinsicht um das Lösen von Problemen. Und bei der Problemlösung in der Informatik dreht sich alles um das Debugging.

Sowohl Lernende als auch Lehrer haben oft Schwierigkeiten, wenn sie von der rein sequentiellen Programmierung zur Programmierung mit dynamischeren Programmierstrukturen wie Schleifen, Verzweigungen und Bedingungen übergehen. Einer der Hauptgründe für diese Schwierigkeiten ist, dass ihnen die Debugging-Fähigkeiten fehlen, die für das erfolgreiche Finden, Identifizieren und Beheben von Fehlern erforderlich sind.

Die Grundlagen des Debuggens werden in dieser Lektion vorgestellt, und die Praxis wird auf die EdScratch-Umgebung angewendet. Diese hauptsächlich Offline-Aktivität wird beginnen, Debugging-Fähigkeiten in Ihren Lernenden zu entwickeln, die entscheidend sind, um sie für den Erfolg bei Aktivitäten in anderen Einheiten einzurichten.

Tipps und Tricks

- Während Technik und Informatik den unglücklichen Ruf entwickelt haben, "für Männer" zu sein, ist die Geschichte der Informatik voller faszinierender Menschen, sowohl männlich als auch weiblich. Grace Murray Hopper (in dieser Aktivität erwähnt), Ada Lovelace, Margaret Hamilton und Hedy Lamarr sind nur einige der fabelhaften Frauen, die dazu beigetragen haben, die moderne Informatik zu dem zu machen, was sie heute ist. Jede dieser Damen ist ein brillantes Thema für Forschungsprojekte und eine wunderbare Inspiration für Lernende jeden Geschlechts.
- Einen vollständigen Leitfaden zu den Warnmeldungen in EdScratch finden Sie unter <https://meetedison.com/robot-programming-software/edscratch/>. Dieser Leitfaden enthält alle Warnmeldungen, was sie bedeuten und Beispiele, wann Sie ihnen begegnen können. Warnmeldungen in EdScratch können Ihnen helfen, jegliche Syntaxfehler zu beheben und geben auch Hinweise, um viele häufige logische Fehler zu beheben.
- Alle Syntaxfehler, die in EdScratch gemacht werden können, erzeugen eine rote Warnmeldung in der Bug-Box. Nicht alle roten Meldungen sind jedoch Syntaxfehler.

- Wenn ein Programm läuft, aber nicht das tut, was Sie erwartet haben, besteht die Möglichkeit, dass ein logischer Fehler im Programm vorliegt. Gewöhnen Sie sich an, die Schülerinnen und Schüler zu fragen, was ihr Programm tun soll, und sie dann das von ihnen geschriebene Programm im gleichen Ablauf wie der Roboter vorlesen zu lassen. Dadurch wird oft der logische Fehler aufgedeckt.

Lösungsschlüssel

Frage	Typ	Answer	Hinweis
1	EA	<p><i>Syntax error: Der Tempoblock darf nicht innerhalb des Gruppierungsblocks "Musik im Hintergrund abspielen" liegen. Nur Musiknotenblöcke können in diesen Block gehen.</i></p> <p><i>Logischer Fehler: das Problem liegt in der vom Programmierer verwendeten sequentiellen Logik. Der Programmierer musste den Tempoblock verwenden, um das Tempo vor den Musikblöcken einzustellen, damit die Musik im sehr schnellen Tempo abgespielt werden konnte.</i></p>	Jede Beschreibung, die die beiden Fehler richtig identifiziert und erklärt, ist akzeptabel.
2	EA	<p><i>Damit das Programm wie beschrieben funktioniert, müssen die folgenden Änderungen vorgenommen werden:</i></p> <p><i>1 - Die Entfernungseingabe im Block "Spin rechts" muss 360 und nicht 630 betragen. [logischer Fehler]</i></p> <p><i>2 - die Geschwindigkeitseingaben für die Blöcke "Spin links" und "Spin rechts" müssen den gleichen Wert haben. [logischer Fehler]</i></p> <p><i>3 - Der Block 'Tempo festlegen' muss außerhalb des Gruppierungsblocks 'Musik im Hintergrund abspielen' liegen [Syntaxfehler].</i></p> <p><i>4 - der Block 'Tempo festlegen' muss vor dem Block 'Musik im Hintergrund abspielen' stehen [logischer Fehler].</i></p> <p><i>5 - die Blöcke 'Drehung links' und 'Drehung rechts' müssen nach dem Block 'Musik im Hintergrund abspielen' stehen [logischer Fehler].</i></p>	Die Antworten der Lernenden sollten alle fünf Fehler identifizieren und beheben. Die Angabe des Fehlertyps wird im Arbeitsblatt nicht verlangt, kann aber bei der Überprüfung der Fehler mit den Lernenden hilfreich sein.

Aktivität U2-2.5 Erforschen wir die Motoren von Edison

Umfang der Tätigkeit	Groß
Empfehlung	
Benötigte Materialien	Basissatz Verbrauchsmaterial, Arbeitsblatt U2-2.5

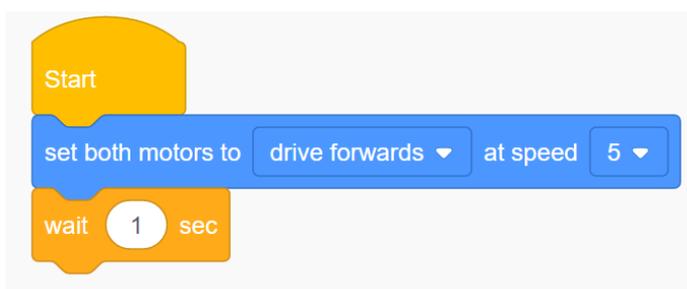
Überblick

Diese Lektion fasst viele der Schlüsselkonzepte aus Einheit 2 zusammen, darunter sequenzielle Programmierung, Ein- und Ausgänge, Eingabeparameter und Debugging. Die Schülerinnen und Schüler erforschen weiter, wie die Motorausgänge von Edison durch Antriebsblöcke gesteuert werden, und lernen, dass die Motoren unabhängig voneinander arbeiten und gesteuert werden können. Sie erhalten eine kreative Denksportaufgabe, um sich eine Verwendung für diese unabhängige Motorbewegung in einer Roboteranwendung vorzustellen.

Bei dieser Aktivität wenden die Lernenden auch ihr Verständnis dafür an, wie Eingaben (einschließlich der Informationseingabe durch Eingabeparameter) die Ausgaben beeinflussen. Sie setzen dieses Verständnis ein, um zu erforschen, warum einige Blöcke in der Sprache von anderen Blöcken abhängig sind. Sie werden dann aufgefordert, ihr Wissen anzuwenden, um ein Programm in einer Aktivität zum Laufen zu bringen, die darauf ausgerichtet ist, sie ein wenig aus ihrer Komfortzone zu bringen.

Tipps und Tricks

- Die Blöcke "Motoren einstellen" schalten nur die Motoren ein. Diese Blöcke benötigen zusätzliche Blöcke, um die Motoraktivität zu steuern, indem eine Bedingung für den Block festgelegt wird. Dieses Programm sagt zum Beispiel, dass die Motoren eingeschaltet und auf Vorwärtsbewegung eingestellt werden sollen, und sagt dann, dass in diesem Zustand 1 Sekunde gewartet werden soll.



- Die "Set-Motors"-Blöcke können anfangs etwas knifflig zu verwenden sein, aber sie sind tatsächlich sehr leistungsfähige Blöcke und sehr hilfreich bei vielen verschiedenen Robotik-Bauprojekten mit Edison.

- Auch wenn Sie nicht vorhaben, die "Set-Motor"-Blöcke in anderen Aktivitäten in dieser Lektion zu verwenden, ist diese Unterrichtsaktivität eine gute Gelegenheit für die Schülerinnen und Schüler, Schwierigkeiten beim Programmieren zu erfahren und zu überwinden.

Lösungsschlüssel

Frage	Typ	Antwort/Beispielantwort	Hinweis
1	EA	<i>Das linke Rad bewegt sich rückwärts.</i>	
2	EA	<i>Das rechte Rad bewegt sich vorwärts.</i>	

3	SE	<p><i>Ich denke, dass die Blöcke "Motor links einstellen" und "Motor rechts einstellen" die Blöcke sind, die nur einen Motor steuern, weil sie sagen, welcher Motor betroffen ist.</i></p>	<p>Die Lernenden wählen vielleicht nicht die richtigen Blöcke aus, aber sie sollten einen Grund haben, die von ihnen getroffene Wahl zu unterstützen. Die Teilnehmer haben vielleicht entdeckt, dass die Eingabeparameter für den "Stopp"-Block auch so eingestellt werden können, dass jeweils nur ein Motor gleichzeitig betroffen ist.</p>															
4	SE	<p><i>Ich würde Edison in einen Ventilator verwandeln, und der eine Motor würde die Ventilatorblätter drehen.</i></p>	<p>Dies ist eine Übung in der Anwendung des kreativen Denkens. Jede Idee, die einen der Motoren von Edison verwendet, ist akzeptabel.</p>															
5	EA	<table border="0"> <thead> <tr> <th>Informationen</th> <th>im Programm?</th> <th>Wert</th> </tr> </thead> <tbody> <tr> <td>Richtung</td> <td>ja</td> <td>vorwärts</td> </tr> <tr> <td>Entfernung</td> <td>ja</td> <td>3</td> </tr> <tr> <td>Abstandseinheiten</td> <td>ja</td> <td>cm</td> </tr> <tr> <td>Geschwindigkeit</td> <td>ja</td> <td>4</td> </tr> </tbody> </table>	Informationen	im Programm?	Wert	Richtung	ja	vorwärts	Entfernung	ja	3	Abstandseinheiten	ja	cm	Geschwindigkeit	ja	4	
Informationen	im Programm?	Wert																
Richtung	ja	vorwärts																
Entfernung	ja	3																
Abstandseinheiten	ja	cm																
Geschwindigkeit	ja	4																
6	EA	<p><i>Der Roboter fuhr mit einer Geschwindigkeit von 4 cm 3 cm vorwärts.</i></p>																
7	EA	<table border="0"> <thead> <tr> <th>Informationen</th> <th>im Programm?</th> <th>Wert</th> </tr> </thead> <tbody> <tr> <td>Richtung</td> <td>ja</td> <td>vorwärts</td> </tr> <tr> <td>Entfernung</td> <td>nein</td> <td></td> </tr> <tr> <td>Entfernungseinheiten</td> <td>nein</td> <td></td> </tr> <tr> <td>Geschwindigkeit</td> <td>ja</td> <td>5</td> </tr> </tbody> </table>	Informationen	im Programm?	Wert	Richtung	ja	vorwärts	Entfernung	nein		Entfernungseinheiten	nein		Geschwindigkeit	ja	5	
Informationen	im Programm?	Wert																
Richtung	ja	vorwärts																
Entfernung	nein																	
Entfernungseinheiten	nein																	
Geschwindigkeit	ja	5																

8	SE	<p><i>Der Roboter hat nichts getan. Er konnte es nicht, weil er nicht alle Informationen hatte, die er brauchte.</i></p>	<p>Der Roboter verfügt nicht über die Informationen, um zu wissen, wie weit er sich bewegen soll, so dass er scheinbar nichts tut. Lernenden, die antworten, werden möglicherweise feststellen, dass in der Bug-Box eine Meldung erscheint, die sie darauf hinweist, dass sie zusätzliche Blöcke benötigen.</p>
9	SE	<p><i>Zuerst versuchte ich, dem Programmstart einen Block 'Musiktempo' hinzuzufügen, aber er bewegte sich immer noch nicht. Ich verschob den 'Tempo'-Block an das Ende des Programms, aber das funktionierte immer noch nicht. Ich schaute mir die anderen Drive-Blöcke an und sah, dass Sekunden eine der Entfernungseinheiten war. Das ließ mich an den "Warte"-Block denken, also fügte ich diesen am Ende des Programms hinzu, und es funktionierte!</i></p> <p><i>Ich war wirklich frustriert, als ich das Programm nicht zum Funktionieren bringen konnte, aber als ich es einmal herausgefunden hatte, war ich wirklich glücklich!</i></p>	<p>Der Schlüssel dazu ist die Entwicklung von Widerstandsfähigkeit. Es ist zweitrangig, welchen Weg die Lernenden eingeschlagen haben und ob sie das Programm zum Funktionieren gebracht haben oder nicht.</p>

Activity U2-2.5a : Spinning garden

Umfang der Tätigkeit	Projekt
Empfehlung	- Vollständige Aktivität U2-2,5 vor dieser Aktivität - Diese Aktivität und die Aktivität U2-2.5b sind beides angewandte Robotik-Projekte. Ziehen Sie in Betracht, mindestens eines davon als Robotik-Bauprojekt für diese Einheit zu verwenden.
Benötigte Materialien	Basisset, Arbeitsblatt U2-2.5a, Makerspace/Handwerkerbedarf, EdCreate-Sätze/LEGO-Steine

Überblick

Dieses offene Projekt lässt die Lernenden das, was sie über sequentielle Programmierung, Ein- und Ausgaben und Debugging gelernt haben, in einem physikalischen Computerprojekt anwenden. Dieses Robotik-Bauprojekt fordert die Schülerinnen und Schüler auf, an der Erstellung eines Spinning Garden mitzuarbeiten. Die Schülerinnen und Schüler müssen ihren Garten planen, die verschiedenen Elemente bauen, die an den Robotern befestigt werden sollen und jeden Roboter so programmieren, dass er seine Befestigung spinnt.

Tipps und Tricks

- Sie können eine Version eines Spinning-Garden-Projekts mit Raspberry-Pi-Boards auf YouTube unter <https://youtu.be/4Fs7y7qZlag?t=424> sehen (Springen Sie zur 7:00-Minuten-Marke, um den endgültigen Spinning-Build in Aktion zu sehen).
- Teil der Idee dieses Projekts ist es, die Roboter auf die Seite zu drehen und die Robotermotoren auf eine andere Art und Weise als zum "Fahren" zu verwenden. Bei diesem Projekt geht es jedoch auch darum, das kreative Denken und die Problemlösung der Schülerinnen und Schüler freizusetzen. Wenn die Schülerinnen und Schüler beschließen, die Roboter auf andere Weise als mit den Robotern auf der Seite zu benutzen, sie aber dennoch zur Gestaltung eines Spinning Garden einsetzen, ist das in Ordnung!

Aktivität U2-2.5b: Sich drehendes Sonnensystem

Umfang der Tätigkeit	Projekt
Empfehlung	- Vollständige Aktivität U2-2,5 vor dieser Aktivität - Diese Aktivität und die Aktivität U2-2.5a sind beides angewandte Robotik-Projekte. Ziehen Sie in Betracht, mindestens eines davon als Robotik-Bauprojekt für diese Einheit zu verwenden.
Benötigte Materialien	Basisset, Arbeitsblatt U2-2.5b, Raum- und Bastelbedarf, EdCreate-Sätze/LEGO-Steine

Überblick

Bei diesem Projekt wenden Lernende das, was sie über sequentielle Programmierung, Ein- und Ausgaben und Debugging gelernt haben, in einem physikalischen Computerprojekt an. Dieses Robotik-Bauprojekt fordert die Schülerinnen und Schüler auf, an der Erstellung eines Modellsonnensystems mitzuarbeiten. Die Schülerinnen und Schüler müssen ihren Ansatz planen, die verschiedenen Elemente bauen, die an den Robotern befestigt werden sollen, und jeden Roboter so programmieren, dass er seine Befestigung dreht.

Tipps und Tricks

- Wenn Sie möchten, dass Ihr Sonnensystem ein genaueres Modell wird, arbeiten Sie mit den Schülerinnen und Schülern zusammen, um die verschiedenen Maßstäbe zu erstellen, die für die Entwicklung eines repräsentativeren Modells erforderlich sind. Bestimmen Sie z.B., in welchem Maßstab die Planeten bemessen sind und welchen Maßstab für die Entfernung, die die einzelnen Planeten voneinander und von der Sonne haben müssen.
- Wenn Sie mehr Kreativität zulassen wollen, lassen Sie die Lernenden ihr eigenes Sternensystem entwerfen, anstatt ein Modell des Sonnensystems zu bauen. Die Schülerinnen und Schüler können ihre eigenen Planeten entwerfen und benennen und über die Zusammensetzung jedes Planeten entscheiden, einschließlich Merkmalen wie Abmessungen, Farbgebung, Bewohnbarkeit usw. Lassen Sie die Schülerinnen und Schüler Geschichten über dieses Sternensystem und alle Bewohnerinnen und Bewohner, die sich darin aufhalten, schreiben.

Aktivität U2-2.5c : Kartograph und Navigator

Umfang der Tätigkeit	Projekt
Empfehlung	- Empfohlen als Höhepunkt nach Abschluss der Aktivitäten von Einheit 2 im Rahmen von Let's Explore - Sowohl diese Aktivität als auch die Aktivität U2-2,5d sind so konzipiert, dass sie als Eckpfeiler der Projektoptionen für Block 2 verwendet werden können. Ziehen Sie in Betracht, mindestens eines davon als Abschlussprojekt für diese Einheit zu verwenden.
Benötigte Materialien	Basisset, Arbeitsblatt U2-2.5c, Makerspace/Zubehör für die Herstellung der Karte

Überblick

Bringen Sie Ihre Schülerinnen und Schüler dazu, die Sequenz auf ein Projekt anzuwenden, bei dem sie den Aufbau, die Herausforderungen und die Lösungen entwerfen. Die Lernenden erstellen eine Karte, die groß genug ist, um sie mit ihren Edison-Robotern zu verwenden, und entwerfen dann Programmieraufgaben, die mit dieser Karte gelöst werden sollen. Nachdem sie ihre Herausforderungen getestet haben um sicherzustellen, dass eine Lösung möglich ist, können sie sich dann mit einem Klassenkameraden austauschen.

Das Entwerfen der Programmierherausforderungen und das Testen ihrer Realisierbarkeit bringt die Schülerinnen und Schüler dazu, mit der Sequenz und den anderen Konzepten aus dieser Einheit auf eine neue Art und Weise zu arbeiten. Dieses Projekt kann als "Schaufenster" für die Schülerinnen und Schüler genutzt werden, um ihre Lernerfahrungen aus der Einheit zu demonstrieren.

Tipps und Tricks

- Dieses Projekt funktioniert gut in Gruppen oder sogar als ganze Klasse. Die Schülerinnen und Schüler können unter Verwendung derselben oder verschiedener Karten Programmieraufgaben miteinander austauschen.
- Ziehen Sie in Erwägung, dieses Projekt mit allen relevanten Geographie- oder Weltstudien der Schülerinnen und Schüler zu verbinden, indem Sie die Schülerinnen und Schüler Karten von Orten anfertigen lassen, über die sie in anderen Fächern lernen.
- Wenn die Schülerinnen und Schüler ihre Karte mit Hilfe eines Rastersystems erstellen, ist sie leichter zu navigieren, aber möglicherweise nicht so genau. Dies ist ein gutes Beispiel für Kompromisse bei der Gestaltung und ein gutes Beispiel für die Diskussion des Themas.

- Hat Ihre Klasse eine "Partnerklasse" mit jüngeren Lernenden? Probieren Sie diese Variante aus: Lassen Sie die älteren Schülerinnen und Schüler eine Karte mit einem Gitternetzsystem und verschiedene Fahraufgaben erstellen. (Vielleicht finden Sie es am besten, wenn Sie es einfach halten und bei diesen Herausforderungen 'Sound'- und 'LED'-Elemente überspringen). Die "Kartographen" können dann mit den jüngeren Lernenden in der Rolle von Mentoren zusammenarbeiten und den anderen Lernenden bei der Lösung der Herausforderungen helfen. Die jüngeren Schüler können entweder einen Druckknopf-Roboter benutzen, Edison mit der EdBlocks-Programmiersprache programmieren oder den älteren Schülern bei der Programmierung des Roboters mit EdScratch helfen. Wie auch immer Sie es machen, wenn Sie Ihren Schülern die Chance geben, das Konzept der Sequenz zu lehren, wird dieses Schlüsselkonzept des rechnerischen Denkens für sie zementiert.

Aktivität U2-2.5d: Schriftsteller und Regisseur

Umfang der Tätigkeit	Projekt
Empfehlung	- Empfohlen als Höhepunkt nach Abschluss der Aktivitäten von Einheit 2 im Rahmen von Let's Explore - Sowohl diese Aktivität als auch die Aktivität U2-2.5c sind so konzipiert, dass sie als Eckpfeiler der Projektoptionen für Block 2 verwendet werden können. Ziehen Sie in Betracht, mindestens eines davon als Abschlussprojekt für diese Einheit zu verwenden.
Benötigte Materialien	Grundbedarfssatz, Arbeitsblatt U2-2.5d, Makerspace /Baumaterial für die Herstellung der Story Map

Überblick

Story Maps sind hilfreiche Werkzeuge sowohl für das Schreiben von Geschichten als auch für die Aufschlüsselung der gelesenen Geschichten, um Verständnis zu zeigen. Story Maps sind großartige Beispiele für Sequenzen in der realen Welt, und diese Verbindung ist die Grundlage für dieses unbefristete Grundsteinprojekt.

Die Schülerinnen und Schüler erstellen eine Geschichte mit Hilfe einer Story Map. Sie "illustrieren" die Geschichte mit Edison - indem sie den Roboter so programmieren, dass er die Geschichte zum Leben erweckt! Die Schülerinnen und Schüler sollten eine Kombination der Edison-Ausgaben unter Verwendung der Motoren, LEDs und Geräusche des Roboters verwenden, um Schlüsselmomente in ihrer Geschichte hervorzuheben, Licht- oder Soundeffekte hinzuzufügen oder wie auch immer sie es wünschen.

Tipps und Tricks

- Eine Story-Map im Stil eines "Märchenberges" eignet sich gut als Basisdesign für dieses Projekt, da sie einen schönen linearen Verlauf schafft. Ein Beispiel für diese Art von Layout können Sie hier sehen: <https://www.tes.com/lessons/wlmfQ5vQh-8OPA/sequencing>
- Dieses Projekt eignet sich gut für Gruppenarbeit.
- Dieses Projekt eignet sich gut für gemischte Medien, indem Präsentationen gefilmt werden - eine großartige "Vorzeige"-Möglichkeit.

Einheit 3: Schleifen programmieren

Erforschen Sie in dieser Einheit das Schlüsselkonzept der Schleifen, indem Sie verschiedene Möglichkeiten untersuchen, wie Schleifen zur Steuerung des Verhaltens der Edison-Roboter verwendet werden können. Das Thema der Programmierlogik wird genauer untersucht, einschließlich der Frage, wie Kontrollstrukturen den Codefluss beeinflussen können. Die Schülerinnen und Schüler entdecken weiterhin die EdScratch-Umgebung, lernen mehr über die verschiedenen Blocktypen in EdScratch und die Art und Weise, wie diese Blöcke zur Erstellung mit Edison verwendet werden können.

Lernziele

Die Schülerinnen und Schüler werden:

- durch die Struktur der Computerprogrammierung von Schleifen in das Konzept des rechnergestützten Denkens der Wiederholung eingeführt werden
- ein Arbeitsverständnis des Unterschieds zwischen bestimmten und unbestimmten Schleifen entwickeln
- die Konzepte von Sequenz und Schleifen in EdScratch zu Arbeitsprogrammen kombinieren
- in das Programmierkonzept der Unterbrechungen eingeführt werden
- Experimentieren Sie mit neuen Blöcken in den Kategorien 'Kontrolle', 'Kommentare' und 'Ereignisse' in EdScratch
- ihre Problemlösungsfähigkeiten durch die Erstellung und Programmierung von Robotik-Projekten erweitern

Kernkompetenzen: Schleifen (definitiv und unbestimmt), Wiederholung, Programmierung von Kontrollstrukturen, Logik, Ereignisse, Unterbrechungen, Kommentare

Lektionen und Aktivitäten in dieser Einheit

Diese Einheit umfasst zwei Lektionen mit insgesamt sechs Basisaktivitäten und 14 Erweiterungsaktivitäten.

Lektion 1: Schleifen

- [U3-1.1 Wir untersuchen sich wiederholende Schritte](#)
 - o [U3-1.1a : Ein Dreieck fahren](#)
 - o [U3-1.1b : Ein Sechseck abfahren](#)
 - o [U3-1.1c Herausforderung angenommen: Wählen Sie Ihre Form](#)
 - o [U3-1.1d : Einen Kreis fahren](#)

- o [U3-1.1e : Ein Quadrat fahren?](#)
- o [U3-1.1f : Doodle-bot-Herausforderung](#)
- [U3-1.2 Untersuchen wir Schleifen und Sequenzen](#)
- [U3-1.3 Lasst uns ewige Schleifen erforschen](#)
 - o [U3-1.3a Herausforderung angenommen: Ohrwurm](#)
- [U3-1.4 Untersuchen wir Stapel- und Verschachtelungsschleifen](#)
 - o [U3-1.4a : Edison, der Entwerfer](#)
 - o [U3-1.4b Herausforderung angenommen: Tanzparty!](#)
- [Lektion 2: Unterbrechungen](#)
 - o [U3-2.1 Lasst uns die Unterbrechung des Hauptprogramms untersuchen](#)
 - o [U3-2.1a : Versuchen Sie stattdessen einen Klatscher](#)
 - o [U3-2.1b Herausforderung angenommen: Betrüger-Bot](#)
 - o [U3-2.1c Herausforderung angenommen: Wählen Sie aus](#)
- [U3-2.2 Wir untersuchen Kommentare in der Kodierung](#)
 - o [U3-2.2a Herausforderung angenommen: Erstellen und kommentieren](#)
 - o [U3-2.2b Herausforderung angenommen: Teilen Sie Ihre Kommentare](#)

Lektion 1: Schleifen

Das Konzept der Schleife eines Programms steht im Mittelpunkt dieser Lektion. Die Schülerinnen und Schüler erforschen die Wiederholung beim Programmieren und die Idee, Codes durch die Verwendung von sogenannten Schleifen effizienter zu machen. Sowohl bestimmte als auch unbestimmte Schleifen werden vorgestellt und anhand einiger der Schleifenblöcke im Abschnitt "Kontrolle" von EdScratch untersucht. Die Schülerinnen und Schüler verbessern ihr Verständnis der Programmierlogik, indem sie sequentielle Programmierung mit Schleifenstrukturen auf verschiedene Weise kombinieren, um mit EdScratch Programme für Edison zu erstellen.

Diese Lektion hat insgesamt vier Basisaktivitäten und neun Erweiterungsaktivitäten:

- [U3-1.1 Wir untersuchen sich wiederholende Schritte](#)
 - o [U3-1.1a : Ein Dreieck abfahren](#)
 - o [U3-1.1b : Ein Sechseck abfahren](#)
 - o [U3-1.1c Herausforderung angenommen: Wählen Sie Ihre Form](#)
 - o [U3-1.1d : Einen Kreis fahren](#)
 - o [U3-1.1e : Ein Quadrat fahren](#)
 - o [U3-1.1f : Doodle-bot-Herausforderung](#)
- [U3-1.2 Untersuchen wir Schleifen und Sequenzen](#)
- [U3-1.3 Unendliche Schleifen untersuchen](#)
 - o [U3-1.3a Herausforderung angenommen: Ohrwurm](#)
- [U3-1.4 Wir untersuchen Stapel- und Verschachtelungsschleifen](#)
 - o [U3-1.4a : Edison, der Entwerfer](#)
 - o [U3-1.4b Herausforderung angenommen: Tanzparty!](#)

Aktivität U3-1.1 Wir untersuchen sich wiederholende Schritte

Umfang der Tätigkeit	Mittel
Empfehlung	Erwägen Sie, diese Aktivität zusammen mit U3-1.1a und/oder U3-1.1b als eine einzige Lektion durchzuführen
Benötigte Materialien	Basisset, Arbeitsblatt U3-1.1, Tätigkeitsblatt U3-1

Überblick

Schleifen, die erste Programmierkontrollstruktur dieser Lektion, werden in dieser Aktivität besprochen. Die Schülerinnen und Schüler schreiben zunächst ein Programm, um Edison in einem Viereck fahren zu lassen, wobei sie nur sequenzielle Programmierung verwenden und das sich wiederholende Muster beachten. Dann werden bestimmte Schleifen eingeführt, und die Schüler erstellen ihr "Drive in a square"-Programm mit dieser effizienteren Programmiermethode neu.

Tipps und Tricks

- Wenn die Schülerinnen und Schüler Schwierigkeiten haben, das Konzept der Schleife zu verstehen, kann es hilfreich sein, auf Papier oder an der Tafel festzuhalten, was in diesem Programm geschieht. Schreiben Sie die "Fahr-" und "Drehbefehle" in aufeinanderfolgender Reihenfolge auf und integrieren Sie sie in eine Schleife. Schreiben Sie die Zahlen 1, 2, 3 und 4 neben die Schleife. Zeigen Sie mit Pfeilen an, dass der Roboter die beiden Befehle der Reihe nach ausführt, und gehen Sie dann zurück an den Anfang der Schleife. Streichen Sie die Zahl 1 durch und wiederholen Sie dann den Vorgang, bis Sie sich durch alle vier Wiederholungen der Schleife bewegt haben.
- Schleifen machen die Programmierung effizienter, aber das bedeutet nicht, dass Programme, die keine Schleifen verwenden, falsch sind. Sowohl das Programm mit acht Blöcken als auch das Programm mit drei Blöcken zu programmieren und dabei zu beobachten, dass beide Programme gleich gut funktionieren, ist eine gute Möglichkeit für die Schülerinnen und Schüler, für sich selbst die Erfahrung zu machen, dass "beim Programmieren mehrere Lösungen möglich sind".
- Indem die Schüler zuerst das Acht-Block-Programm und dann das Drei-Block-Programm schreiben, können sie sehen, wie Programmstrukturen die Erstellung von Programmen einfacher und effizienter machen können.
- Wenn die Räder von Edison den Rand des Papiers des Aktivitätsblatts erwischen, kann dies den Roboter leicht aus der Bahn werfen. Sie können dies beheben, indem Sie das Arbeitsblatt mit Klebeband festkleben oder das Muster des Arbeitsblatts auf einem größeren Blatt Papier nachbilden.

- Aufgrund geringfügiger mechanischer Unterschiede zwischen den Motoren und Encodern in den verschiedenen Edison-Robotern drehen sich einige Roboter bei einer Eingabe von 90 Grad möglicherweise nicht genau um 90 Grad. Ermutigen Sie die Schülerinnen und Schüler, verschiedene Werte um 90 auszuprobieren (z.B. 87 oder 93), um die Eingabe zu finden, die für ihren Edison am besten funktioniert.

- Bei der Verwendung von Programmen, die mehrere "Fahrbefehle" nacheinander ausführen, können die Schülerinnen und Schüler feststellen, dass die Genauigkeit ihres Edison-Roboters abnimmt, wenn immer mehr Befehle ausgeführt werden. Durch das Hinzufügen einer Pause (entweder durch eine "Wartezeit" mit einem sehr kurzen Eingabewert oder gegebenenfalls durch einen "Stopp-Motoren"-Block) zwischen den "Fahrbefehlen" können die Motoren die Bewegung vollständig stoppen, was die Genauigkeit erhöht.

Lösungsschlüssel

Frage	Typ	Antwort/Beispielantwort	Hinweis
1	EA	8	Wenn die Schülerinnen und Schüler nur sieben Blöcke haben, beendet ihr Roboter wahrscheinlich nicht die letzte Runde, wodurch er wieder in die Ausgangsposition zurückkehrt, wie auf dem Arbeitsblatt angegeben.
2	SE	<i>Die Blöcke haben ein Muster. Es werden nur zwei Blocktypen mit den gleichen Eingabeparametern verwendet, und sie wiederholen sich viermal.</i>	Die Schüler sollten ein Wiederholungsmuster bezüglich der verwendeten Blöcke und/oder der Reihenfolge der Blöcke im Programm beachten.
3	EA	4	

4	SE	<i>Der Wert muss 4 sein, da ein Quadrat vier Seiten und vier Ecken hat. Das Programm muss also auf jeder Seite fahren und jede Ecke insgesamt 4 Mal umdrehen.</i>	Im Idealfall stellen die SchülerInnen eine Korrelation zwischen der Anzahl der erforderlichen Schleifenaktionen und den Seiten/Winkeln der Form fest.
---	----	---	---

Aktivität U3-1.1a : Ein Dreieck fahren

Umfang der Tätigkeit	Klein
Empfehlung	- Vollständige Aktivität U3-1.1 vor dieser Aktivität - Erwägen Sie, diese Aktivität zusammen mit U3-1.1 und/oder U3-1.1b als eine einzige Lektion durchzuführen
Benötigte Materialien	Set Basismaterial, Arbeitsblatt U3-1.1a, Tätigkeitsblatt U3-2

Überblick

Die Schüler üben die Verwendung der definitiven Schleifenprogrammierstruktur, um ein effizientes Programm zu schreiben, das Edison dazu bringt, ein Dreieck zu fahren. Diese Aktivität soll in Verbindung mit der Aktivität U3-1.1 verwendet werden, um Schülern zu helfen, den Zusammenhang zwischen der Anzahl der Schleifen in einer definitiven Schleife und der resultierenden Ausgabe zu beherrschen.

Tipps und Tricks

- Die Berechnung des Winkels in Grad, den der Roboter drehen muss, um das Dreieck abzutasten, kann knifflig sein. Erinnern Sie die Schüler daran, dass die Summe der Innenwinkel eines Dreiecks 180° beträgt und dass das Dreieck auf dem Arbeitsblatt ein gleichseitiges Dreieck ist.
- Diese Aktivität eignet sich gut für eine Geometrieerweiterungsstunde.
- Wenn die Räder von Edison den Rand des Papiers des Arbeitsblatts erwischen, kann dies den Roboter leicht aus der Bahn werfen. Sie können dies beheben, indem Sie das Arbeitsblatt mit Klebeband festkleben oder das Muster des Arbeitsblatts auf einem größeren Stück Papier nachbilden.

Lösungsschlüssel

Frage	Typ	Antwort/Beispielantwort	Hinweis
1	SE	3	Die Mindestanzahl der Blöcke, die für ein erfolgreiches Programm benötigt werden, beträgt 3: eine Schleife, ein Vorwärtsfahrblock und ein Wendeblock.
2	EA	3	
3	SE	<i>Der Wert muss 3 sein, da ein Dreieck drei Seiten und drei Ecken hat. Der Code muss also jede Seite fahren und jede Ecke insgesamt 3 Mal umdrehen.</i>	Im Idealfall stellen die SchülerInnen eine Korrelation zwischen der Anzahl der erforderlichen Schleifenaktionen und den Seiten/Winkeln der Form fest.

Aktivität U3-1.1b : Einen Sechskant antreiben

Umfang der Tätigkeit	Klein
Empfehlung	- Vollständige Aktivität U3-1.1 vor dieser Aktivität - Erwägen Sie, diese Aktivität zusammen mit U3-1.1 und/oder U3-1.1a als eine einzige Lektion durchzuführen
Benötigte Materialien	Basissatz, Arbeitsblatt U3-1.1b, Tätigkeitsblatt U3-3

Überblick

Schüler üben die Verwendung der definitiven Schleifenprogrammierstruktur, um ein effizientes Programm zu schreiben, das Edison dazu bringt, ein Sechseck zu fahren. Diese Aktivität soll in Verbindung mit der Aktivität U3-1.1 verwendet werden, um Schülern zu helfen, den Zusammenhang zwischen der Anzahl der Schleifen in einer definitiven Schleife und der resultierenden Ausgabe zu beherrschen.

Tipps und Tricks

- Es kann schwierig sein, den Winkel in Grad zu berechnen, den der Roboter drehen muss, um das Sechseck abzutasten. Erinnern Sie die Schüler daran, dass die Summe der Innenwinkel eines Sechsecks 720° beträgt und dass das Sechseck auf dem Arbeitsblatt eine regelmäßige (d.h. alle Seiten sind gleich) 6-seitige Form hat.
- Diese Aktivität eignet sich gut für eine Geometrieerweiterungsstunde.
- Wenn die Räder von Edison den Rand des Papiers des Arbeitsblatts erwischen, kann dies den Roboter leicht aus der Bahn werfen. Sie können dies beheben, indem Sie das Arbeitsblatt mit Klebeband festkleben oder das Muster des Arbeitsblatts auf einem größeren Stück Papier nachbilden.

Lösungsschlüssel

Frage	Typ	Antwort/Beispielantwort	Hinweis
1	SE	3	Die Mindestanzahl der Blöcke, die für ein erfolgreiches Programm benötigt werden, beträgt 3: eine Schleife, ein Vorwärtsfahrtblock und ein Wendeblock.
2	EA	6	

3	SE	<i>Der Wert muss 6 sein, da ein Sechseck sechs Seiten und sechs Ecken hat. Der Code muss also jede Seite fahren und jede Ecke insgesamt 6 Mal umdrehen.</i>	Die Schüler werden idealerweise eine Korrelation zwischen der Anzahl der erforderlichen Schleifenaktionen und den Seiten/Winkeln der Form feststellen.
---	----	---	--

Aktivität U3-1.1c Herausforderung angenommen: Wählen Sie Ihre Form

Umfang der Tätigkeit	Mittel
Empfehlung	Vollständige Aktivität U3-1.1 und entweder U3-1.1a oder U3-1.1b vor dieser Aktivität
Benötigte Materialien	Grundversorgungssatz, Arbeitsblatt U3-1.1c, Zubehör zur Herstellung eigener Formen

Überblick

Die Schüler wenden ihr Verständnis der Beziehung zwischen einer bestimmten Schleife und einer regelmäßigen Form an, indem sie eine Form auswählen und dann ein effizientes Programm schreiben, das Edison dazu bringt, diese Form zu fahren. Bei dieser Aktivität werden die Schüler aufgefordert, die Muster zu beschreiben, die ihnen bei der Verwendung bestimmter Schleifen mit Formen aufgefallen sind und daraus zu folgern, um andere Formen zu erforschen.

Tipps und Tricks

- Um eine Form mit Hilfe eines 3-Block-Programms zu vervollständigen, das aus einer definitiven Schleife besteht, die einen "Antriebs"-Block und einen "Dreh"-Block enthält, müssen die Schüler eine reguläre Form auswählen. Wenn Schüler sich jedoch entscheiden, keine reguläre Form zu verwenden, können sie die Herausforderung trotzdem versuchen. Sie können die Aktivität U3-1.2 zuerst ausführen, wenn die Schüler nicht-regelmäßige Formen verwenden möchten.

Lösungsschlüssel

Frage	Typ	Antwort/Beispielantwort	Hinweis
1	EA	12	
2	SE	<i>Meine Form war ein Achteck, das acht Seiten und Winkel hat. Der Eingabeparameter muss immer gleich der Anzahl der sich wiederholenden Seiten und Winkel sein, die die Form hat, daher wusste ich, dass ich eine 8 als Eingabeparameter des Wiederholungsblocks benötigen würde.</i>	Die Schüler werden idealerweise einen Zusammenhang zwischen der Anzahl der erforderlichen Schleifenaktionen und den Seiten/Winkeln der Form feststellen.

Aktivität U3-1.1d : Einen Kreis fahren

Umfang der Tätigkeit	Klein
Empfehlung	Vollständige Aktivität U3-1.1 und entweder U3-1.1a oder U3-1.1b vor dieser Aktivität
Benötigte Materialien	Basisset, Arbeitsblatt U3-1.1d, Tätigkeitsblatt U3-4

Überblick

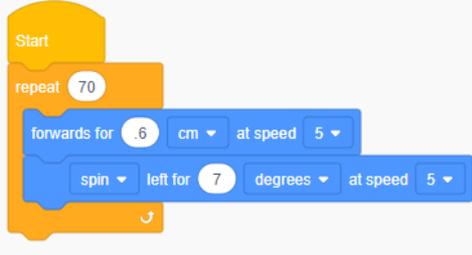
Diese Aktivität ist eine großartige Übung in kreativer Problemlösung und eine Erforschung der Grenzen der definitiven Schleifenprogrammierungsstruktur.

Die Schüler wenden bei dieser Herausforderung ihr Verständnis der Beziehung zwischen einer bestimmten Schleife und einer regelmäßigen Form an, indem sie anhand des von ihnen beobachteten Musters extrapolieren, um ein Programm zu erstellen, in dem Edison im Kreis fährt. Diese Aktivität soll eingesetzt werden, sobald die Schülerinnen und Schüler erkannt haben, dass die Anzahl der Schleifen, die erforderlich ist, um den Roboter dazu zu bringen, eine Form zu fahren, gleich der Anzahl der Seiten dieser Form ist. Um Edison im Kreis zu fahren, müssen die Schüler die Idee erweitern und ein wenig kreativ werden.

Tipps und Tricks

- Wenn Schüler feststecken, können Sie ihnen den Hinweis geben, dass "eine Form mit vielen sehr niedrigen Seiten einem Kreis sehr nahe kommen kann".
- Der Eingabeparameter im Wiederholungsblock hat eine Wertobergrenze von 1000. Wenn Schüler eine größere Anzahl eingeben, wird der Block automatisch auf 1000 begrenzt.
- Wenn die Räder von Edison den Rand des Papiers des Aktivitätsblatts erwischen, kann dies den Roboter leicht aus der Bahn werfen. Sie können dies beheben, indem Sie das Arbeitsblatt mit Klebeband festkleben oder das Muster des Arbeitsblatts auf einem größeren Blatt Papier nachbilden.

Lösungsschlüssel

Frage	Typ	Antwort/Beispielantwort	Hinweis
1	RC		
2	SE	<p><i>Nein, Edison fährt eigentlich nicht im Kreis, sondern in einer Form mit vielen Seiten. Es sieht aus wie ein Kreis, aber es ist kein richtiger Kreis.</i></p>	<p>Schüler können den Roboter möglicherweise dazu bringen, einem Kreis sehr nahe zu kommen. Der Roboter fährt jedoch nicht in einem durchgehenden Bogen, sondern macht Niedrig-Vorwärtsbewegungen und dreht sich in jeder Schleife leicht.</p>

Aktivität U3-1.1e : Ein Quadrat fahren?

Umfang der Tätigkeit	Klein
Empfehlung	Vollständige Aktivität U3-1.1 vor dieser Aktivität
Benötigte Materialien	Basisset, Arbeitsblatt U3-1.1e, Tätigkeitsblatt U3-1

Überblick

Erhöhen Sie den Schwierigkeitsgrad, indem Sie den Zufallszahleneingabeblock einführen. Diese Aktivität fügt der definitiven Schleife etwas Abwechslung hinzu und ist eine nette Art und Weise, Mathematik in der Programmierung zu präsentieren. Die Konzepte werden zwar nicht explizit vorgestellt, aber die Zufallszahleneingabe wird im Speicher von Edison als variabler Wert gespeichert. Variablen, die in Einheit 5 dieser Lektionen vorgestellt werden, werden bei der Verwendung von Edison-Sensoren in Programmen ausgiebig verwendet.

Tipps und Tricks

- Aufgrund geringfügiger mechanischer Unterschiede bei den Motoren und Encodern in den verschiedenen Edison-Robotern müssen die Schüler möglicherweise die Eingabeparameter der Antriebsblöcke in diesem Programm an ihre Roboter anpassen. Um die Genauigkeit zu verbessern, können Schüler auch einen "Warte"-Block mit einem Sortier-Eingangswert (z.B. 2 Sekunden) zwischen den Antriebsbefehlen hinzufügen.
- Sie können die Schüler auffordern, den Programmablauf auf dem Arbeitsblatt zu beobachten und die Anzahl der Seiten zu zählen, die der Roboter antreibt. Daraus können sie die Anzahl der Schleifen ermitteln, die das Programm bei diesem Lauf durchlaufen hat.
- Auch wenn die Anzahl der Schleifen, die das Programm durchläuft, zufällig ist, handelt es sich dennoch um eine definitive Schleife. Das Programm wird eine bestimmte Anzahl von Schleifen durchlaufen, die einem Wert aus der Menge von (1, 2, 3, 4, 5, 6, 7, 8, 9, 10) entspricht.
- Diese Aktivität eignet sich gut für eine Wahrscheinlichkeitserweiterungsaktivität.
- Wenn die Räder von Edison den Rand des Papiers des Aktivitätsblatts erwischen, kann dies den Roboter leicht aus der Bahn werfen. Sie können dies beheben, indem Sie das Aktivitätsblatt mit Klebeband festkleben oder das Muster des Aktivitätsblattes auf einem größeren Stück Papier nachbilden.

Lösungsschlüssel

Frage	Typ	Antwort/Beispielantwort	Hinweis
1	SE	<i>Als ich das Programm zum ersten Mal ausführte, fuhr der Roboter eine ganze Zeit lang um den Platz herum, plus 3 weitere Seiten, so dass er 7 Mal eine Schleife fuhr. Beim zweiten Mal schaffte er nur eine halbe Runde um das Quadrat, so dass er nur 2 Mal eine Schleife fuhr! Jedes Mal passieren andere Dinge, weil die Anzahl der Schleifen zufällig ist, nicht immer 4.</i>	

Aktivität U3-1.1f : Doodle-bot-Herausforderung

Umfang der Tätigkeit	Groß
Empfehlung	
Benötigte Materialien	Basisset, Arbeitsblatt U3-1.1f, EdCreate-Sätze/LEGO-Steine und/oder andere Materialien für das Makerspace, Filzstifte oder Ähnliches

Überblick

Diese Herausforderung fügt der Aufgabe, den Roboter dazu zu bringen, so zu fahren, dass eine Form entsteht, ein konstruktives Element hinzu. Als eine Herausforderung mit offenem Ende bietet diese Aktivität den Schülern sehr wenig Anleitung oder Grenzen. Die Schüler wenden ihr Verständnis der sequentiellen Programmierung und der definitiven Regelkreisstruktur an, um den Roboter dazu zu bringen, in einer Form zu fahren. Die Hauptherausforderung dieser Aktivität besteht darin, eine Möglichkeit zu entwerfen und zu bauen, einen Stift am Roboter anzubringen, so dass diese Form auf Papier nachgezeichnet werden kann.

Tipps und Tricks

- Es ist eine Herausforderung, ein Design zu entwickeln, das genügend Druck nach unten auf den Stift ausübt, damit er das Papier markieren kann, während der Roboter immer noch fahren kann. Durch die Verwendung von Filzstiften oder einem anderen Schreibgerät, das leicht Tinte hinterlässt, lässt sich diese Herausforderung leichter bewältigen.
- Diese Herausforderung kann als eine konstruktive Herausforderung angesehen werden. Sie können mehr über die Verwendung dieser Herausforderung für die Lehre des technischen Designs unter <https://meetiedison.com/teach-engineering-design-with-edcreate/> lesen.

Lösungsschlüssel

Frage	Typ	Antwort/Beispielantwort	Hinweis
1	SE	<p><i>Es war schwierig, den Stift an den Roboter zu bekommen. Mein erster Plan funktionierte überhaupt nicht, und ich musste von vorne anfangen. Mein zweiter Plan benutzte Gummibänder, um den Stift an der Seite des Roboters anzubringen, und das funktionierte. Ich programmierte den Roboter so, dass er ein Achteck zeichnet, aber als ich ihn laufen ließ, waren die Gummibänder über dem Rad, was bedeutete, dass Edison nicht fahren konnte. Schließlich befestigte ich den Stift mit Hilfe einiger Baubalken und Gummibänder an der Rückseite von Edison und brachte den Roboter zum Fahren. Edison fuhr in der Achteckform, zeichnete ihn aber nicht sehr gut, weil nicht genug Druck auf den Stift ausgeübt wurde, um ihn gegen das Papier zu drücken.</i></p>	<p>Die Antworten der Schülerinnen und Schüler sollten einige der Herausforderungen oder Probleme abdecken, mit denen sie konfrontiert waren, und wie sie mit diesen Problemen umgegangen sind.</p>

Aktivität U3-1.2 Wir untersuchen Schleifen und Sequenzen

Umfang der Tätigkeit	Mittel
Empfehlung	
Benötigte Materialien	Basissatz, Arbeitsblatt U3-1.2, Tätigkeitsblatt U3-5

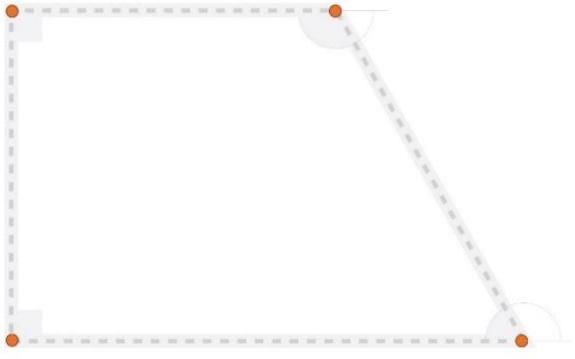
Überblick

Diese Aktivität lässt Schüler ein effizientes Programm schreiben, das Edison dazu bringt, eine unregelmäßige Form zu fahren. Die Schüler müssen sowohl ihre Beherrschung des Konzepts der Sequenz als auch ihr Verständnis der definitiven Schleifenstruktur einsetzen, um diese Aufgabe zu bewältigen. Die Verwendung eines Vierecks als die Form, die Edison fahren soll, ist eine gute Möglichkeit, die Grenzen von Schleifen zu betrachten und die Bedeutung der Sequenz beim Programmieren zu verstärken. Durch die Wahl des richtigen Startpunktes auf einem Viereck kann eine bestimmte Schleife trotzdem verwendet werden und macht das Programm effizienter. Das Programm benötigt jedoch zusätzliche Sätze außerhalb der Schleife, um die Form zu vervollständigen.

Tipps und Tricks

- Wenn die Schülerinnen und Schüler Schwierigkeiten haben zu bestimmen, was innerhalb und was außerhalb der Schleife liegen soll, arbeiten sie die Form des Vierecks mit Ihren Schülerinnen und Schülern durch. Schauen Sie sich an, was sich in der Form wiederholt und was nicht. Bestimmen Sie auf diese Weise, welche Schritte innerhalb der Schleife liegen können und in welcher Reihenfolge der gesamte Code im Programm benötigt wird.
- Wenn die Räder von Edison den Rand des Papiers des Aktivitätsblatts erwischen, kann dies den Roboter leicht aus der Bahn werfen. Sie können dies beheben, indem Sie das Arbeitsblatt mit Klebeband festkleben oder das Muster des Arbeitsblatts auf einem größeren Stück Papier nachbilden.

Lösungsschlüssel

Frage	Typ	Antwort/Beispielantwort	Hinweis
1	SE		
2	SE	<p><i>Ich startete den Roboter an der Ecke mit dem stumpfen Winkel und zeigte auf die Ecke mit dem rechten Winkel. Das bedeutete, dass die ersten beiden Seiten, die der Roboter zum Fahren brauchte, gleich lang waren und die ersten beiden Drehungen beide 90 Grad waren. Aus diesem Grund ist der erste Teil meines Codes eine Schleife, die sich zweimal wiederholt. Der Rest des Programms besteht aus einzelnen Blöcken außerhalb einer Schleife, da sich die anderen Seiten und Winkel auf der Form nicht wiederholen und daher nicht mit einer Schleife im Programm verwendet werden können.</i></p>	<p>Schüler sollten bei ihren Antworten eine Korrelation zwischen dem von ihnen verwendeten Startpunkt und der Schleifenbildung des Programms beachten. NB: Einige Startpunkte erlauben es dem Programm nicht, eine Schleife zu verwenden.</p>

Aktivität U3-1.3 Lasst uns ewige Schleifen erforschen

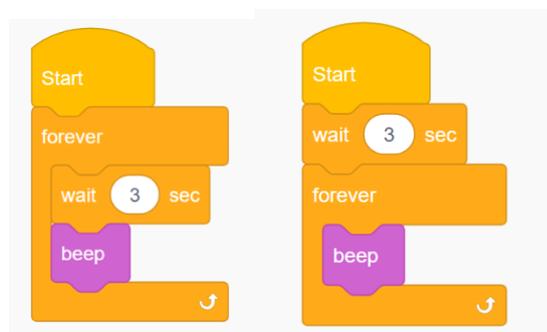
Umfang der Tätigkeit	Klein
Empfehlung	Es wird empfohlen, vor dieser Aktivität bestimmte Schleifen einzuführen.
Benötigte Materialien	Basissatz, Arbeitsblatt U3-1.3

Überblick

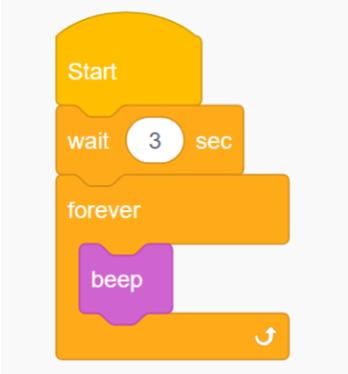
Diese Aktivität führt den zweiten Haupttyp der Schleifenstruktur ein: die unbestimmte Schleife. Während bestimmte Schleifen im Allgemeinen leichter zu verstehen sind als unbestimmte Schleifen, haben unbestimmte Schleifen einen viel größeren Anwendungsbereich in der Computerprogrammierung. Die "ewige" Schleife, die eine unbestimmte Schleife ist, wird in dieser Aktivität eingeführt und erforscht. Die Schüler verwenden diesen neuen Block in der halboffenen Programmieraufgabe "Ei-Timer", wodurch sie ihr Verständnis von Sequenz- und Schleifenstrukturen anwenden können.

Tipps und Tricks

- Jedes Programm kann jederzeit durch Drücken der Stop-Taste (Quadrat) gestoppt werden. Dadurch wird eine "ewige" Schleife beendet und der Roboter wieder in den Standby-Modus versetzt.
- Vielleicht möchten Sie die Schülerinnen und Schüler ermutigen zu testen, was passiert, wenn sie sowohl einen Warte- als auch einen Piep-Block innerhalb der "ewigen" Schleife setzen, im Vergleich dazu, wenn der Warte-Block außerhalb und der Piep-Block innerhalb der Schleife ist. Dies ist ein großartiges Beispiel für einen logischen Fehler:



Lösungsschlüssel

Frage	Typ	Antwort /Beispielantwort	Hinweis
1	SE	<i>Ich glaube nicht, dass Sie Edison nach der Ewigkeitsschleife noch etwas anderes tun lassen können. Zum einen endet die "Unendlichkeit" nicht, also hört diese Schleife mit "never stop repeating" nicht auf. Zum anderen unterscheidet sich die Form der Schleife von anderen Blöcken - der Boden ist quadratisch, so dass es keine Möglichkeit gibt, einen nächsten Step-Block anzuhängen.</i>	Im Idealfall notieren die Schüler in ihren Antworten die Funktion, Form oder beides des Blocks als Unterstützung für ihre Antwort.
1	RC		

Aktivität U3-1.3a: Ohrwurm

Umfang der Tätigkeit	Klein
Empfehlung	Beenden Sie die Aktivität U3-1.2 vor dieser Aktivität
Benötigte Materialien	- Basissatz, Arbeitsblatt U3-1.3a - Optional: Noten oder Zugang zum Nachschlagen von Noten

Überblick

Diese Aktivität bietet Schülern eine weitere Möglichkeit, die "ewige" unbestimmte Schleifenstruktur auf einfache, aber sinnvolle Weise zu nutzen. Die Verwendung der "ewigen" Schleife zur Wiederholung von Notenblöcken ist eine gute Möglichkeit, das in der realen Welt häufig vorkommende Phänomen darzustellen, wie ein Lied in unseren Köpfen hängen bleibt - z.B. ein Ohrwurm. Das Üben der Verwendung von Programmierstrukturen mit Verbindungen aus der realen Welt hilft den Schülern, die Struktur mit ihrem Zweck in Verbindung zu bringen und ermöglicht es ihnen, die Verwendung dieser Strukturen in späteren Aktivitäten zu erkennen, wenn sie komplexere Programmieraufgaben in Angriff nehmen.

Tipps und Tricks

- Wenn Sie möchten, dass Schüler mit der "ewigen" Schleife üben, aber keine Zeit mit der Musikauswahl verbringen möchten, können Sie die Schüler bitten, das Demoprogramm `Moving_with_music` zu öffnen und das Programm so zu ändern, dass ein Ohrwurmprogramm erstellt wird, das die Melodie des Demoprogramms als Lied verwendet.
- Um zu betonen, wo die Schleife endet, schlagen Sie den Schülern vor, einen "Warte"-Block als letzten Block innerhalb ihres Ohrwurmprogramms hinzuzufügen, so dass Edison eine Pause macht, bevor das Lied erneut beginnt.

Aktivität U3-1.4 Wir untersuchen Stapel- und Verschachtelungsschleifen

Umfang der Tätigkeit	Mittel
Empfehlung	
Benötigte Materialien	Basissatz, Arbeitsblatt U3-1.4, Tätigkeitsblatt U3-6

Überblick

Schüler werden in das Konzept der Verwendung mehrerer Schleifen in einem Programm eingeführt, sowohl durch sequentielles Stapeln als auch durch Verschachtelung einer Schleife in eine andere. Das Verschachteln von Schleifen mag wie eine neue Art der Manipulation des Programmablaufs erscheinen, aber in Wirklichkeit handelt es sich nur um eine Schichtfolge. Die Arbeit mit Programmen mit gestapelten und verschachtelten Schleifen ermöglicht es Schülern, die rechnerischen Denkpraktiken der Zerlegung in kleinere Teile und Mustererkennung sowie die Anwendung sequentieller Logik anzuwenden.

Tipps und Tricks

- Wenn Schüler Schwierigkeiten haben, das Schleifenkonzept insbesondere ineinander verschachtelter Schleifen zu verstehen, kann es hilfreich sein, das Geschehen auf Papier oder an der Tafel festzuhalten. Sie können dies tun, indem Sie mit Ihren Schülern nacheinander jeden Befehl aufschreiben oder indem Sie das Programm so aufschreiben wie es erscheint, und dann mit Pfeilen auf jede Schleifenaktion hinweisen.
- Die Aktivität in Aufgabe 1 (sich den Code ansehen und herausfinden, was er tut, ohne das Programm laufen zu lassen) ähnelt einer Codierungspraxis, die als Tracing bekannt ist. Bei der formalen Praxis der Ablaufverfolgung wird ein Programm Zeile für Zeile durchlaufen und wichtige Werte aufgezeichnet. Dies wird oft getan, um Fehler oder Bugs im Code zu finden, aber es ist auch nützlich, wenn Sie nur verstehen müssen, was in einem Programm geschieht. Obwohl das Konzept in dieser Aktivität nicht explizit vorgestellt wird und die Notwendigkeit, Werte in EdScratch aufzuzeichnen, begrenzt ist, wird es den Schülern helfen, ihre Fähigkeit, Programme zu debuggen, zu verbessern, wenn sie sich daran gewöhnen, durch Programme zu gehen.
- Wenn die Räder von Edison den Rand des Papiers des Aktivitätsblatts erwischen, kann dies den Roboter leicht aus der Bahn werfen. Sie können dies beheben, indem Sie das Arbeitsblatt mit Klebeband festkleben oder das Muster des Arbeitsblatts auf einem größeren Stück Papier nachbilden.

Lösungsschlüssel

Frage	Typ	Antwort/Beispielantwort	Hinweis
1	EA	3	
2	EA	<i>piepst</i>	
3	EA	15	
4	EA	<i>LED blinkt</i>	
5	SE	<i>Es sind drei Quadrate in einer Reihe.</i>	
6	SE	<i>Ich denke, ich kann eine verschachtelte Schleife verwenden, um Edison dazu zu bringen, ein Viereck zu fahren (das eine Schleife verwendet, um 4 Mal zu fahren und sich 4 Mal zu drehen), und diese Schleife dann in einer anderen Schleife zu haben, die sich 3 Mal wiederholt.</i>	Im Idealfall sehen Schüler eine Korrelation zwischen dem sich wiederholenden Muster und der verschachtelten Schleife.

Aktivität U3-1.4a : Edison der Entwerfer

Umfang der Tätigkeit	Groß
Empfehlung	- Empfohlen zur Verstärkung verschachtelter Schleifen - Sehr empfehlenswert, wenn Schüler mit Schleifenstrukturen zu kämpfen haben
Benötigte Materialien	Basissatz, Arbeitsblatt U3-1.4a, Tätigkeitsblatt U3-7, Bordvorräte zur Herstellung des Versuchsraums

Überblick

Geben Sie Schülern die Möglichkeit, Schleifen und Sequenzen auf die Probe zu stellen, indem Sie Edison in fabelhaften Designs fahren lassen. Bei dieser Aktivität sehen sich Schüler Entwürfe an, um sich wiederholende Muster zu identifizieren, und konstruieren dann so effektiv wie möglich eine Code-Lösung unter Verwendung von Schleifen und verschachtelten Schleifen.

Tipps und Tricks

- Die physische Erstellung des Designs durch Markieren oder Zeichnen kann Schülern helfen, die Wiederholungen, die innerhalb des Musters auftreten, zu

erkennen und diese besser mit den Schleifenstrukturen zu verbinden, die sie innerhalb ihrer EdScratch-Codelösung benötigen.

- Schülerinnen und Schüler ihre eigenen Muster erstellen zu lassen, ist eine großartige Möglichkeit, sie kritisch darüber nachdenken zu lassen, wie Wiederholungen in der Praxis aussehen. Ein Design zu erstellen, das ein Muster innerhalb eines Musters hat, ist eine großartige Gelegenheit, rechnergestütztes Denken auf das Design anzuwenden.

Aktivität U3-1.4b : Tanzparty!

Umfang der Tätigkeit	Groß
Empfehlung	
Benötigte Materialien	- Basissatz, Arbeitsblatt U3-1.4b - Optional: Geräte zum Abspielen von Musik

Überblick

Bringen Sie die Roboter in Bewegung, indem Sie Edisons Ausgänge und die Regelkreisstrukturen in Aktion setzen. In diesem Miniprojekt wenden Schüler ihr Verständnis von Sequenz, Wiederholung und Problemlösung in einer gemeinschaftlichen Umgebung an.

Tipps und Tricks

- Diese Aktivität ist für den Einsatz in Gruppen gedacht, um Schülern die Zusammenarbeit an einem Projekt zu ermöglichen. Schüler können über jeden Schritt der Entwurfs-, Kodierungs- und Testphase hinweg zusammenarbeiten. Alternativ können die Schüler die Arbeit mit Teammitgliedern, die für bestimmte Aufgaben zuständig sind, aufteilen.
- Der Hokey Pokey ist eine großartige Songauswahl, wenn die Schüler keine Inspiration finden können. Der Liedtext gibt einen Plan vor, wozu Schüler die Roboter programmieren können!

Lesson 2: Unterbrechungen

Erweitern Sie Ihr Verständnis der Berechnungslogik, indem Sie sich mit Unterbrechungen in der Programmierung beschäftigen. Diese Lektion führt neue Blöcke aus der Kategorie 'Ereignisse' in EdScratch ein und verwendet diese Blöcke zur Erstellung von Unterprogrammen. Schüler lernen, wie durch Unterbrechungen ausgelöste Unterprogramme funktionieren und wenden ihr Verständnis von

Sequenzen an, um Programmen zu folgen, die diese Art von Unterprogrammen enthalten. Die Computerprogrammierpraxis des Kommentierens wird auch als ein Werkzeug vorgestellt, das Schülern hilft, ihren Code und ihre Logik zu verfolgen und zu erklären.

Diese Lektion umfasst insgesamt zwei Basisaktivitäten und fünf Erweiterungsaktivitäten:

- [U3-2.1 Lasst uns die Unterbrechung des Hauptprogramms untersuchen](#)
 - o [U3-2.1a Ändern Sie es nach oben: Versuchen Sie stattdessen einen Klatscher](#)
 - o [U3-2.1b Herausforderung angenommen: Betrüger-Bot](#)
 - o [U3-2.1c Herausforderung angenommen: Wähle eins aus](#)
- [U3-2.2 Wir untersuchen Kommentare in der Kodierung](#)
 - o [U3-2.2a Herausforderung angenommen: Erstellen und kommentieren](#)
 - o [U3-2.2b Herausforderung angenommen: Teilen Sie Ihre Kommentare](#)

Aktivität U3-2.1 Unterbrechen wir das Hauptprogramm

Umfang der Tätigkeit	Mittel
Empfehlung	
Benötigte Materialien	Basissatz, Arbeitsblatt U3-2.1

Überblick

Diese Aktivität führt eine Reihe von neuen Begriffen und Konzepten ein, die sich alle auf eine zentrale Idee beziehen: die Unterbrechung des Hauptprogramms. Mit der Einführung von Unterbrechungen und Unterprogrammen erhält das grundlegende Konzept des rechnergestützten Denkens von Sequenzen eine neue Dimension. Diese wichtigen Programmierkonzepte sind von entscheidender Bedeutung, um die ereignisbasierte Programmierung mit Edison freizuschalten, damit die Leistung der Sensoren des Roboters genutzt werden kann.

Diese Unterrichtsaktivität geht auf Themen ein, die zuvor eingeführt wurden, um Schülern zu helfen zu verstehen, wie ereignisgesteuerte Interrupts funktionieren. Die Schüler werden in die Blockkategorie "Ereignisse" in EdScratch eingeführt und üben

mit einem Programm, das einen Interrupt und eine Unteroutine hat, einen "Entscheidungsroboter" mit ihrem Edison-Roboter zu erstellen.

Tipps und Tricks

- Unterbrechungen unterbrechen zwar den sequentiellen Ablauf eines Computerprogramms, sind aber immer noch an die gleiche zugrunde liegende Logik gebunden. Subroutinen führen die in der Subroutine enthaltenen Befehle in sequentieller Reihenfolge aus. Nach Beendigung kehrt das Programm genau an den Punkt des Hauptprogramms zurück, an dem es sich vor der Unterbrechung befand, und fährt dann mit der Ausführung des Hauptprogramms in sequentieller Reihenfolge fort.
- Der Decider-Bot kann zur Beantwortung jeder binären Frage verwendet werden. Sie können wählen, ob die Schüler ihre eigenen binären Fragen erstellen oder Fragen mit zwei a- oder b-ähnlichen Optionen erstellen, die mit dem Entscheidungs-Bot beantwortet werden.

Lösungsschlüssel

Frage	Typ	Antwort/Beispielantwort	Hinweis
1	EA	<i>Das Hauptprogramm verwendet eine unbestimmte "ewige" Schleife, um den Roboter dazu zu bringen, die rechte LED einzuschalten, 0,03 Sekunden zu warten, die rechte LED auszuschalten, die linke LED einzuschalten, 0,03 Sekunden zu warten und die linke LED auszuschalten. Diese Schleife von Aktionen wird für immer wiederholt.</i>	Schüler können die Antwort anders formulieren, aber sie sollten die Aktionen des Hauptprogramms wie beschrieben identifizieren.
2	SE	<i>Alle Blöcke der Veranstaltungskategorie sind gelb und haben die gleiche Form wie der "Start"-Block.</i>	Schüler sollten Ähnlichkeiten zwischen den Veranstaltungsblöcken und dem Standardblock Start erkennen.

Aktivität U3-2.1a : Klatschsensoren

Umfang der Tätigkeit	Niedrig
Empfehlung	Vollständige Aktivität U3-2.1 vor dieser Aktivität
Benötigte Materialien	- Basissatz, Arbeitsblatt U3-2.1a - Optional: Arbeitsblatt U3-2.1 als Referenz

Überblick

Diese Erweiterungsaktivität ist zur Verwendung in Verbindung mit Aktivität U3-2.1 vorgesehen. Schüler programmieren ihre Edison-Roboter erneut als Entscheidungsroboter. Diesmal wird die Unterbrechung nicht durch einen Knopfdruck verursacht, sondern wenn der Roboter ein Klatschen feststellt.

Während das Hauptziel dieser Aktivität darin besteht, die Konzepte von Unterbrechungen und Unterprogrammen zu verstärken, dient die Aktivität auch als eine gute sanfte Einführung in die Benutzung der Edison-Sensoren. Sie kann als vorbereitende Aktivität verwendet werden, um Schülern zu helfen, sich bei der Programmierung mit Sensoren vertraut zu machen, bevor sie mit den anderen Sensoren von Edison arbeiten.

Tipps und Tricks

- Die Roboter können Schwierigkeiten haben, Geräusche zu erkennen, wenn ein hohes Niveau an Hintergrundgeräuschen vorhanden ist. Wenn Schüler mit dem Finger in die Nähe des Schallsensors ihres Edison tippen, entsteht der gleiche Effekt wie beim Klatschen.
- Das Drücken der "Abspielen"-Taste (Dreieck) zum Ausführen eines Programms in Edison kann einen niedrigen Geräuschpegel erzeugen, der „Klatschen erkannt“ Unterbrechungen auslösen kann. Wenn Schüler direkt nach dem Klatschen erkannt"-Ereignis eine "Wartezeit" haben (wie in diesem Beispielprogramm <https://www.edscratchapp.com?share=pbz13M0a>), kann dies den Anschein erwecken, dass das Programm unmittelbar nach dem Drücken der "Abspielen"-Taste (Dreieck) nicht richtig funktioniert. Tatsächlich hat das Ereignis gerade erst ausgelöst und das "Warten" verursacht. Lassen Sie die Schüler warten, bis die eingestellte Zeit verstrichen ist, dann sehen sie, dass das Programm zum Hauptprogramm zurückspringt und auf den nächsten Ereignisauslöser wartet.

Aktivität U3-2.1b : Betrüger-Bot

Umfang der Tätigkeit	Niedrig
Empfehlung	Vollständige Aktivität U3-2.1 vor dieser Aktivität
Benötigte Materialien	- Basissatz, Arbeitsblatt U3-2.1b - Optional: Arbeitsblatt U3-2.1 als Referenz

Überblick

Diese Erweiterungsaktivität ist zur Verwendung in Verbindung mit Aktivität U3-2.1 vorgesehen. Schüler programmieren ihre Edison-Roboter erneut als Entscheidungsroboter. Diesmal müssen sie zusätzlich zur Standard-Unterroutine 'Pause', die läuft, wenn sie durch eine Unterbrechung von einem der Knöpfe (rund oder dreieckig) ausgelöst wird, eine zweite Unterroutine als 'Cheat' hinzufügen, die eine festgelegte Antwort gibt, sobald der andere Knopf gedrückt wird.

Zusätzlich zur Vertiefung der Konzepte von Unterprogrammen und Unterbrechungen lernen die Schüler, dass sie mehr als ein Unterprogramm pro Programm haben können. Obwohl diese Aktivität nicht explizit vorgestellt wird, demonstriert sie die Verzweigung in der Programmierung und dient als sanfte Einführung in das Konzept der Verzweigung.

Tipps und Tricks

- Ein EdScratch-Programm kann immer nur eine Unterroutine haben, die mit einem bestimmten Ereignis beginnt. Aus diesem Grund verschwindet jeder Ereignisblock, wie z.B. der Ereignisblock 'runde Taste gedrückt', von der Blockpalette, wenn er sich im Programmierbereich befindet.
- Eine mögliche Antwort auf die Aktivität des Cheater-Bot kann unter <https://www.edscratchapp.com?share=K072a7Ym> eingesehen werden.

Activity U3-2.1c : Wähle eins aus

Umfang der Tätigkeit	Groß
Empfehlung	Vollständige Aktivität U3-2.1 vor dieser Aktivität
Benötigte Materialien	- Basissatz, Arbeitsblatt U3-2.1c, Materialien für die Herstellung der Box - Optional: Arbeitsblatt U3-2.1 als Referenz

Überblick

Diese Erweiterungsaktivität ist zur Verwendung in Verbindung mit Aktivität U3-2.1 vorgesehen. Schüler programmieren ihre Edison-Roboter erneut als Entscheidungsroboter. Diese Aktivität fordert die Schüler dann auf, eine physische Box oder ein anderes System zu erstellen, das sie mit ihrem Decider-Bot verwenden können. Dieser physische Entwurf sollte es Edison ermöglichen, eine Auswahl zu "erleuchten". Die Schülerinnen und Schüler könnten zum Beispiel eine Box mit zwei Fenstern entwerfen, in die Wörter oder Bilder eingeschoben werden können. Wenn das Decider-Bot-Programm läuft, wird eine der beiden Optionen von den LEDs beleuchtet.

Diese physische Design-Aktivität verstärkt die Konzepte von Unterprogrammen und Unterbrechungen in der Programmierung und verlangt von den Schülern, sich mit der Problemlösung zu befassen, um einen greifbaren Nutzen für ein solches Programm zu schaffen.

Tipps und Tricks

- Je nach Entwurf müssen Schüler möglicherweise die Wartezeit in ihrem Unterprogramm verlängern.
- Der physikalische Entwurf muss es ermöglichen, dass die Knöpfe des Roboters gedrückt werden können, um die Unterroutine auszulösen. Erinnern Sie die Schüler daran, wenn sie ihre Entwürfe erstellen.

Aktivität U3-2.2 Untersuchen wir die Kommentare in der Codierung

Umfang der Tätigkeit	Mittel
Empfehlung	
Benötigte Materialien	Basissatz, Arbeitsblatt U3-2.2

Überblick

Die Praxis des Kommentierens in der Programmierung und die Blockkategorie "Kommentar" in EdScratch werden in dieser Aktivität vorgestellt. Schüler sehen, wie Kommentare in EdScratch aussehen und verwenden ein Beispielprogramm, um zu verstehen, wie Kommentare ein hilfreiches Werkzeug sind.

Einige professionelle Programme mögen Kommentare nicht, da sie behaupten, dass Kommentare Zeitverschwendung sind und argumentieren, dass "guter Code selbstdokumentierend sein sollte". Gute Kommentare helfen jedoch dabei, großartigen Code zu erstellen, und aus pädagogischer Sicht ist das Kommentieren von Kommentaren eine sehr hilfreiche Praxis. Wenn sich Schüler daran gewöhnen, das, was sie zu erreichen versuchen, zu artikulieren, insbesondere wenn sie mit komplexeren Code-Strukturen arbeiten, können sie beim Debuggen weitaus unabhängiger Problemlöser sein. Das Kommentieren ermöglicht auch eine leichtere Zusammenarbeit, da andere Schüler und Moderatoren die Absichten des Programmierers schneller verstehen können. Darüber hinaus hilft das Kommentieren den Schülern, ihr "Coder-Denken" in menschenlesbaren Begriffen zu erklären. Die Fähigkeit, effektiv über technologische Arbeit zu kommunizieren, ist eine Schlüsselkompetenz in jedem digitalen Alphabetisierungsarsenal.

Tipps und Tricks

- Wenn Schüler Schwierigkeiten haben, zu verstehen, was Kommentare sind, sollten Sie in Erwägung ziehen, Kommentare anhand eines gedruckten Dokuments und einiger Haftnotizen zu erklären. Zeigen Sie, wie Sie Haftnotizen in das Dokument einfügen können, damit andere Personen an bestimmten Stellen "Hinweise" auf den Inhalt des Dokuments erhalten können. Die Haftnotizen sind eine hilfreiche Möglichkeit, Notizen über den "Code" (das Dokument) zu hinterlassen, sind aber nicht wirklich ein Teil des Dokuments.
- Erinnern Sie die Schüler daran, dass Edison die Kommentarblöcke vollständig ignorieren wird. Die Roboter können diese Blöcke nicht verstehen, egal, was in sie hineingelegt wird.
- Das Hinzufügen eines Kommentars, der sagt, was das Programm tun soll, bringt das Programm nicht dazu, diese Sache zu tun: der eigentliche Code muss noch funktionieren.
- Das einfache Hinzufügen von Kommentaren 'debuggt' ein Programm nicht. Kommentare können jedoch ein großartiges Hilfsmittel für Schüler sein, mit dem sie logische Fehler finden und beheben können, das ihnen hilft, ihr eigenes Denken durchzuarbeiten und beim Debuggen Änderungen an ihrem Code vorzunehmen.

- Die Verwendung von Kommentaren in der Codierung ist nicht streng definiert und variiert je nach dem einzelnen Programmierer. Es gibt jedoch ein paar Best-Practice-Regeln, die Sie vielleicht einführen möchten:
 - Kommentare sollten leicht verständlich sein - d.h. sie sollten "menschenslesbar" sein (und nicht nur Codeschnipsel).
 - Kommentare sollten im Allgemeinen am Anfang des Codes, den sie erklären, stehen.
- Kommentare müssen nur dann hinzugefügt werden, wenn der Abschnitt des Codes nicht selbstverständlich ist.

Lösungsschlüssel

Frage	Typ	Antwort/Beispielantwort	Hinweis
1	SE	<i>Ich denke, dass Edison zwischen 1 und 5 Quadrate langsam vorwärts und dann zwischen 1 und 5 Dreiecke schnell rückwärts fahren wird. Edison wird dann diesen ganzen Satz von Aktionen 2 weitere Male wiederholen (insgesamt 3 Mal). Wenn ich zu irgendeinem Zeitpunkt, während das Programm läuft, Beifall bekomme, wird Edison einen Piepston abgeben.</i>	

2	SE	<i>Ja, die Kommentare sind hilfreich, denn bei so vielen Schleifen ist es etwas schwierig zu erkennen, was passieren wird. Mit den Kommentaren ist der Code für die Ansteuerung der Quadrate und Dreiecke mit der Zufallszahlschleife für jedes Quadrat und Dreieck leichter zu verstehen.</i>	Dies ist eine meinungsbasierte Antwort, sollte aber unabhängig von der Meinung des Schülers durch eine solide Argumentation untermauert werden.
3	SE	<i>Das Programm funktionierte größtenteils so, wie ich es erwartet hatte, aber der Roboter piepte die ganze Zeit. Ich dachte nicht, dass er das tun würde, weil ich nicht einmal Beifall klatschte.</i>	Jede Erfahrung eines Schülers ist in Ordnung, aber Sie können prüfen, ob die Antwort des Schülers auf Frage 1 mit der hier notierten Erfahrung übereinstimmt.

4	EA	<i>Die Fehlerbox enthält eine Warnung, die dies erklärt: Das Antreiben der Motoren erzeugt Geräusche, die das Ereignis 'Beifall' auslösen können. Dies kann dazu führen, dass die Blöcke für das 'Beifall'-Ereignis wiederholt ausgelöst werden, während Edison fährt.</i>	Die in der Antwort/Beispielantwort festgestellte Ursache ist der Grund für dieses Auftreten. Schülerantworten müssen diese Ursache notieren.
---	----	--	--

Aktivität U3-2.2a : Erstellen und Kommentieren

Umfang der Tätigkeit	Groß
Empfehlung	Empfohlen als Krönung der Aktivitäten nach Abschluss der "Lasst uns erkunden"-Aktivitäten aus Einheit 3
Benötigte Materialien	- Basissatz, Arbeitsblatt 2.2a - Optional: Materialien für das Makerspace

Überblick

Schülerinnen und Schüler zeigen, was sie in der Einheit gelernt haben, und lassen in diesem ergebnisoffenen Meilensteinprojekt ihr kreatives Denken spielen. Diese Aktivität fordert die Schüler dazu heraus, unter Verwendung der erlernten Programmierstrukturen, Kodierungspraktiken und rechnergestützten Denkfähigkeiten ein selbst entworfenes Programm zu erstellen. Während einige wenige Parameter zur Festlegung von Mindestanforderungen an die Programminhalte vorgegeben sind, steht es den Schülern frei, den Zweck des Programms zu entwerfen.

Tipps und Tricks

- Erinnern Sie die Schüler daran, dass Edison drei Arten von Ausgängen hat: Töne, LEDs und Motorausgänge.
- Sie können sich dafür entscheiden, die Schülerinnen und Schüler dazu zu ermutigen, bei diesem Projekt einen Roboterbau-Ansatz zu verfolgen und ein Programm zu entwerfen, das ihren Roboter als Teil einer technischen Lösung einschließlich einer Art physischer Konstruktion verwendet.
- Anstatt die Schülerinnen und Schüler ihre Programme manuell kopieren zu lassen, können Sie sich dafür entscheiden, dass sie eine Screen-Shot-

Software verwenden, um ein Bild zu erfassen und auszudrucken. Alternativ können Sie die Schüler auffordern, ihr Programm als Datei zu speichern und als Antwort den Speicherort der Datei oder einen Link zu dem Ort, an dem sie es auf einem freigegebenen Laufwerk speichern, anzugeben. Eine gespeicherte Datei ist von Vorteil, wenn Sie die Schüler die Aktivität U3-2.2b durchführen lassen.

Frage	Typ	Antwort/Beispielantwort	Hinweis
1	SE	<i>Ich möchte, dass mein Programm Edison in einen Quizguide Bot verwandelt. Dies wird ein wenig wie ein Entscheidungsroboter sein, aber anstatt eine zufällige Antwort zu geben, möchte ich ihn dazu benutzen, jemandem zu zeigen, ob er eine Frage richtig oder falsch beantwortet hat, wenn ich ihn befrage. Er wird 2 Unterprogrammen haben, eine, die zeigt, dass die Antwort richtig war, und eine, die zeigt, dass sie falsch war. Wenn das Programm läuft, blinken die LEDs von Edison an und aus. Sobald die Person die Frage beantwortet hat, kann ich den Dreiecksknopf drücken, um eine Melodie zu spielen und die Lichter auszuschalten, wenn die Antwort falsch ist, oder den runden Knopf drücken, um eine andere Melodie zu spielen, während sich der Roboter mit eingeschaltetem Licht dreht, wenn die Antwort richtig ist.</i>	Als Referenz zeigt dieser Link das ursprünglich vom Programmierer konzipierte Programm, das in der Antwort/Beispielantwort beschrieben wird.
2	SE	<i>Ich benutzte Kommentare, um die Unterprogramme zu beschriften, so dass ich wusste, welches für die richtige Antwort und welches für die falsche Antwort war. Ein Kommentar befindet sich oben in der Unterroutine des Dreiecksknopfes und sagt 'falsche Antwort'.</i>	Vielleicht möchten Sie bestätigen, dass die Schüler bei der Wahl des Ortes und des Kommentarinhalts gute Kommentarpraktiken anwenden.

3	SE	<p>Ein Problem, das ich hatte, war, dass die Musik noch lief, als die Steuerung der LEDs durch die Subroutine beendet war. Ich habe dies im Unterprogramm für die Dreieckstaste behoben, indem ich den Block "Im Hintergrund abspielen" entfernt habe. Ein weiteres Problem, das ich hatte, war, dass die Subroutine mit den runden Knöpfen den Roboter für immer zum Drehen brachte. Ich behob dies, indem ich den Block in einen Antriebsblock mit einer festgelegten Zeit änderte, anstatt einen Warteblock und den Block 'Motoren einstellen' zu verwenden.*</p>	<p>Jede Erfahrung ist akzeptabel, aber im Allgemeinen werden Schüler auf Probleme stoßen. Wenn ein Schüler festgestellt hat, dass er auf keine Probleme gestoßen ist, sollten Sie sein Programm überprüfen und ihn dazu ermutigen, Risiken einzugehen und kreativ zu denken, wenn das Programm, das er sich ausgedacht hat, sehr einfach ist.</p>
4	RC	<p><i>Ein Musterprogramm ist unter https://www.edscratchapp.com/?share=GDLRNq0X verfügbar.</i></p>	<p>Wenn Schüler einen Link zu ihrer lokal gespeicherten Datei angeben, anstatt ihre Programme auszuschreiben, können sie das Programm wiederverwenden, wenn Sie die Schüler die Aktivität U3-2.2b ausführen lassen.</p>

□

Lösungsschlüssel

* NB: Der Schüler hätte das Problem, dass sich der Roboter für immer dreht, auch dadurch lösen können, dass er am Ende dieses Unterprogramms einen Block "Stopp beider Motoren" hinzugefügt hätte. Dies ist ein gutes Beispiel dafür, dass mehrere Codierungslösungen gleichermaßen geeignet sind, ein Problem zu debuggen.

Aktivität U3-2.2b : Teilen Sie Ihre Kommentare

Umfang der Tätigkeit	Mittel
Empfehlung	
Benötigte Materialien	Basissatz, Arbeitsblatt U3-2.2b

Überblick

Zusammenarbeit ist das Herzstück einer professionellen Programmierung. Nicht nur Teams von Menschen innerhalb einer Organisation kommen bei Programmierprojekten zusammen, auch Einzelpersonen aus der ganzen Welt tauschen online einen Code aus und teilen ihn mit anderen. Gute Kommunikation ist der Schlüssel für eine erfolgreiche Zusammenarbeit bei der Programmierung. Diese Aktivität erforscht genau das und fordert Schülerinnen und Schüler heraus, kritisches Denken zu verwenden, um ihren eigenen Ansatz und den eines Partners zu bewerten.

Tipps und Tricks

- Diese Aktivität kann auch mit Niedrig-Gruppen anstelle von Paaren durchgeführt werden.
- Wenn Schüler ihre Programme aus der Aktivität U3-2.2a abgeschlossen und gespeichert haben, können Sie sie die gespeicherten Dateien verwenden lassen, um Programme auszutauschen und schneller in den zweiten Teil dieser Aktivität zu gelangen.
- Die Schüler müssen ihre Programme für diese Aktivität nicht in den Edison-Robotern laufen lassen, aber es kann hilfreich sein, wenn sie sehen wollen, was ein Programm macht, um über die enthaltenen Kommentare zu entscheiden.
- Vielleicht möchten Sie den Schülern einige Best-Practice-Regeln für professionelle Kommentare zur Verfügung stellen. Alternativ dazu können Sie die Schüler bitten, ihre Antworten mit diesen "bewährten Praktiken" zu vergleichen und eventuelle Unterschiede zu diskutieren:
 - Kommentare sollten leicht verständlich sein - d.h. sie sollten "menschenslesbar" sein (und keine Codeschnipsel darstellen).
 - Kommentare sollten im Allgemeinen am Anfang des Codes, den sie erklären, stehen.

- o Kommentare müssen nur dann hinzugefügt werden, wenn der Abschnitt des Codes nicht selbstverständlich ist.

Lösungsschlüssel

Frage	Typ	Antwort/Beispielantwort	Hinweis
1	SE	<p>1. Kommentare sollten ziemlich kurz gehalten werden</p> <p>2. Kommentare sollten nicht wiederholen, was bereits im Code klar ist</p> <p>3. Kommentare sollten immer in knifflige oder verwirrende Abschnitte eingefügt werden</p> <p>4. Wenn der Kommentar darauf zurückzuführen ist, dass Sie sich über etwas nicht sicher sind (z.B. zur Fehlersuche), fügen Sie ein Fragezeichen in den Kommentar ein</p>	<p>Die Regeln, die sich die Schülerinnen und Schüler ausdenken, sind weit weniger wichtig als der Prozess</p> <p>- Sie sollten sich von den Schülerinnen und Schülern erklären lassen, wie sie ihre Regeln entwickelt haben, um zu verstehen, wie sie zusammengearbeitet haben und welche Logik dahinter steckt.</p>

Lektion 4: Was wäre, wenn...

Erkunden Sie die Auswahl und Verzweigung in der Computerprogrammierung durch die wichtigsten rechnergestützten Konzepte von Bedingungen und Ereignissen. In dieser Einheit werden wichtige Computerprogrammierfähigkeiten, wie z.B. die Entwicklung von Pseudocode, vorgestellt, um den Schülern zu helfen, ihre Problemlösungsfähigkeiten zu verbessern, während sie die verschiedenen Sensorfähigkeiten der Edison-Roboter freischalten. Die Schüler lernen Algorithmen kennen und nutzen dieses Verständnis, um Programme zu erstellen, die ein autonomeres Verhalten der Roboter ermöglichen. Bedingungen, Sensorik, Unterbrechungen und ereignisbasiertes Programmieren werden durch die physikalischen Rechenaktivitäten in dieser Lektion zum Leben erweckt.

Lernziele

Schüler werden:

- in das Konzept des rechnergestützten Denkens der bedingten Selektion (d.h. Verzweigung) eingeführt werden
- untersuchen, wie Eingaben mit Edison funktionieren, indem neue Blöcke in den Kategorien 'Kontrolle', 'Empfinden' und 'Ereignisse' in EdScratch verwendet werden
- experimentieren, unter Verwendung von Edisons Sensoren und Eingabemöglichkeiten in Arbeitsprogrammen
- die Konzepte von Sequenz, Schleifen und Auswahl zu Arbeitsprogrammen in EdScratch kombinieren
- ein praktisches Verständnis dafür entwickeln, wie Pseudocode und Kommentare zur Planung, Verfolgung und Fehlersuche in Programmen verwendet werden können
- in die Idee von Algorithmen in der Computerprogrammierung eingeführt werden und lernen, wie sich Algorithmen von Programmen unterscheiden
- ihr Verständnis von Roboteranwendungen durch Projekte erweitern, bei denen Sensorik und Eingaben verwendet werden

Kernkompetenzen: Bedingungen (d.h. Auswahl und Verzweigung), Ereignisse, Abtastung und Sensoren, Pseudocode, Algorithmen

Lektionen und Aktivitäten in dieser Einheit

Diese Einheit umfasst zwei Lektionen mit insgesamt neun Basisaktivitäten und 15 Erweiterungsaktivitäten.

Lektion 1: Bedingte Bedingungen

- [U4-1.1 Die Verwendung von Konditionalen](#)
 - o [U4-1.1a: Roboterfehler oder menschliches Versagen?](#)
- [U4-1.2 „Wenn“ \(„If“\)- Aussagen](#)
- [U4-1.3 „Wenn“ \(„If“\)-Aussagen und Sequenzen](#)
- [U4-1.4 Untersuchen wir das Stapeln und Verschachteln von if-Anweisungen](#)
 - o [U4-1.4a Herausforderung angenommen: Eine Riemenscheibe bauen](#)

Lektion 2: Fühlen

- [U4-2.1 Pseudocode untersuchen](#)
 - o [U4-2.1a Die Antwort finden](#)
- [U4-2.2 Untersuchen wir Edisons Linienvorfolger](#)
 - o [U4-2.2a Innerhalb einer Grenze fahren](#)
- [U4-2.3 Untersuchen wir Algorithmen](#)
 - o [U4-2.3a: Es gibt mehr als einen Weg, einer Linie zu folgen](#)
- [U4-2.4 Edisons Hinderniserkennung untersuchen](#)
 - o [U4-2.4a Schneller, schneller, zerschlagen?](#)
 - o [U4-2.4b Wenn Linie, nach rechts gehen. Wenn Hindernis, nach links gehen](#)
 - o [U4-2.4c Wo liegt das Hindernis?](#)
 - o [U4-2.4d 3D-Labyrinth](#)
- [U4-2.5 Nachrichtenaustausch mit Edison untersuchen](#)
 - o [U4-2.5a Ferngesteuerte Flaggenmaschine](#)
 - o [U4-2.5b Herausforderung angenommen: Bau und Steuerung des EdCrane](#)
 - o [U4-2.5c Herausforderung angenommen: Wasserwerfer zur Brandbekämpfung](#)
 - o [U4-2.5d Herausforderung oben: Halbautomatischer Bagger](#)

- o [U4-2.5e Herausforderung angenommen: Entfernung von gefährlichem Material](#)
- o [U4-2.5f Herausforderung angenommen: Brieftauben](#)

Lektion 1: Bedingungen

Zusammen mit Schleifen gehören Konditionale zu den wichtigsten Bausteinen, die zur Erstellung von sinnvollem Code in jeder Computerprogrammiersprache benötigt werden. In dieser Lektion wird dieses grundlegende Konzept des rechnergestützten Denkens eingeführt, das als "bedingte Auswahl" oder "Verzweigung" bekannt ist. Die Entwicklung eines fundierten Verständnisses von Konditionalen ist entscheidend für den Erfolg in der zweiten Lektion der Einheit, die in Edisons Sensoren eintaucht.

Die Schüler entwickeln ein Verständnis dafür, wie sequentielle Programmierung und Programmierlogik in Programmen funktionieren, die bedingte Strukturen verwenden. Bedingte Strukturen machen mehrere Optionen innerhalb eines Programms möglich - je nachdem, ob die Bedingung erfüllt ist oder nicht, werden unterschiedliche Dinge geschehen. In dieser Lektion werden sowohl "bis"-Bedingungsstrukturen und "wenn"-Strukturen als auch die diesen Strukturen entsprechenden EdScratch-Blöcke eingeführt. Schüler erstellen Programme in EdScratch unter Verwendung von zuvor erforschten Bedingungsangaben (insbesondere Tastendruck und Beifall) mit sowohl 'bis'- als auch 'wenn'-Bedingungsstrukturen, um zu untersuchen, wie Bedingungen in der Praxis funktionieren.

Diese Lektion umfasst insgesamt vier Basisaktivitäten und zwei Erweiterungsaktivitäten:

- [U4-1.1 Die Verwendung von Konditionalen untersuchen](#)
 - o [U4-1.1a Roboterfehler oder menschliches Versagen?](#)
- [U4-1.2 „Wenn“ - Ansagen](#)
- [U4-1.3 „Wenn“ \(„If“\)-Aussagen und Sequenzen](#)
- [U4-1.4 Wir untersuchen das Stapeln und Verschachteln von if-Anweisungen](#)
 - o [U4-1.4a Herausforderung angenommen: Eine Riemenscheibe bauen](#)

Aktivität U4-1.1 Wir untersuchen die Verwendung von Konditionalen

Umfang der Tätigkeit	Groß
Empfehlung	
Benötigte Materialien	Basissatz, Arbeitsblatt U4-1.1

Überblick

In dieser Aktivität werden Conditionals, einer der grundlegendsten Bausteine, die erforderlich sind, um mit dem Schreiben sinnvoller Programme in jeder Computersprache beginnen zu können, eingeführt. Conditionals im Code sind genau wie Conditionals im Leben: eine Sache ist davon abhängig, dass eine andere Sache zuerst geschieht.

Obwohl "Wenn"-Aussagen die Art von bedingter Struktur sind, die am häufigsten in der Codierung verwendet werden, werden in dieser Aktivität die "bis"-Blöcke in EdScratch verwendet, um das Konzept der Konditionalen einzuführen. Die "bis"-Blöcke werden verwendet, weil die Idee einer Aktion, die "bis" eine bestimmte Bedingung erfüllt ist, für Schülerinnen und Schüler anfangs leichter zu verstehen ist.

Das Arbeitsblatt führt die Schüler zunächst mit Hilfe einer unbestimmten "Wiederholung bis" Schleife durch ein Programm, die als perfekter Ausgangspunkt für das Verständnis des Gesamtkonzepts der Bedingungen dient. Die Schüler verwenden dann verschiedene Beispielprogramme, um das Konzept weiter zu erforschen, und üben dabei die Anwendung der rautenförmigen Ereignisbedingungen in verschiedenen "bis"-Blöcken. Die letzte Frage im Arbeitsblatt fordert die Schülerinnen und Schüler auf zu versuchen, eine mögliche Verwendung einer "Bis-Bedingung"-Struktur in der Technologie, die sie in ihrem täglichen Leben sehen, zu identifizieren, um ihnen zu helfen, das Gelernte mit der Programmierung, die uns in der realen Welt umgibt, zu verbinden.

Tipps und Tricks

- Wenn Schüler sich mit dem Konzept der Bedingtheiten auseinandersetzen, arbeiten sie Beispiele aus dem täglichen Leben durch, die Bedingtheiten verwenden. Zum Beispiel kann die Aufgabe des Geschirrspülens als "Geschirr spülen, bis alles sauber ist" und das Einräumen der Wäsche als "Wäsche zusammenlegen, bis der Wäschekorb leer ist" betrachtet werden.
- Die Verwendung von Programmen, die Tastendrucke als Ereignisse innerhalb des Programms haben, kann verwirrend sein. Erinnern Sie die Schüler daran, dass sie die "Abspielen"-Taste (Dreieck) einmal drücken müssen, um ein Programm auszuführen. Dieses erste Drücken startet NUR das Programm - es wird nicht als ein Knopfdruck innerhalb des Programms betrachtet. Mit

anderen Worten, sobald das Programm läuft, sieht der Roboter den aktuellen Zustand als "Dreieckstaste gedrückt? - NEIN' AN.

- Die Syntax von EdScratch verwendet 'bis' anstelle von 'während' (was in vielen Allzweck-Codiersprachen üblich ist), aber die Code-Struktur führt die gleiche bedingte Funktion aus. Wenn Sie EdScratch zusammen mit einer anderen Sprache unterrichten, die 'while conditionals' verwendet, sollten Sie diese Verbindung den Schülern erklären und darauf hinweisen, dass es keinen funktionellen Unterschied zwischen 'while condition = true' und 'until condition = false' gibt.
 - Beispiel: (während $x < 4$) dasselbe ist wie (bis $x = 4$). In beiden Fällen bricht die Bedingung ab, wenn $x = 4$.

Lösungsschlüssel

Frage	Typ	Antwort/Beispielantwort	Hinweis
1	EA	<i>Die Kategorien "Operatoren" und "Ereignisse" haben Blöcke, die meiner Meinung nach als bedingte Eingabeparameter verwendet werden können. Ich denke das, weil sie rautenförmige Blöcke haben und die 'bis'-Blöcke rautenförmige Löcher haben.</i>	Schüler sollten bei ihren Antworten mindestens die Korrelation der Rautenform beachten.
2	EA	<i>Sie müssen den 'runden' Knopf drücken, während das Programm läuft. Dadurch wird die Bedingung für die Schleife 'Wiederholen bis' gestellt und die Schleife beendet. Das Programm geht dann zum nächsten Satz in der Reihenfolge über, d.h. zum "Piep"-Satz.</i>	Die Antworten der Schüler sollten erkennen lassen, dass der runde Knopfdruck die Bedingung erfüllt und dass das Programm dann in normaler Reihenfolge fortgesetzt wird.

3	EA	4	
---	----	---	--

4	SE	<i>Ein Beispiel aus dem wirklichen Leben für ein Programm, das gut mit einem Block "Warten bis Zustand" beginnen könnte, ist ein Einbruchalarm. Die Ereignisbedingung wäre ein Bewegungserkennungsereignis. Das Programm würde dem Alarm sagen, er solle warten, bis eine Bewegung erkannt wurde. Wenn dann eine Bewegung erkannt wird, würde es einen Alarm auslösen.</i>	Diese Frage drängt Schüler dazu, zu versuchen, die Programmkonzepte der Ereignisbedingungen und der "bis"-Bedingungsstruktur mit der sie umgebenden Welt zu verknüpfen. Es wird nicht erwartet, dass sie die vorhandene Technologie genau erklären, sondern nur, dass sie versuchen, ihr Verständnis auf die Technik anzuwenden.
---	----	--	--

Aktivität U4-1.1a : Roboterfehler oder menschliches Versagen?

Umfang der Tätigkeit	Mittel
Empfehlung	<ul style="list-style-type: none"> - Beenden Sie die Aktivität U4-1.1 vor dieser Aktivität - Empfehlung, das Debugging mit Conditionals zu üben - Stark zu empfehlen, wenn Schüler mit Konditionalitäten zu kämpfen haben
Benötigte Materialien	Basissatz, Arbeitsblatt U4-1.1a

Überblick

Schüler arbeiten ein Programm durch, das Sequenz, Schleifen und Bedingungen verwendet, um herauszufinden, warum sich der Roboter nicht so verhält, wie der Programmierer es erwartet. Dies mag zwar nur wie eine reine Debugging-Aktivität erscheinen, in Wirklichkeit geht es aber mehr darum, rechnergestütztes Denken in Aktion zu betrachten. Die Aktivität führt Schüler durch eine Übung zur Zerlegung eines Programms und seiner "Probleme" in kleinere Einheiten. Die Schüler arbeiten sich dann durch die jedem Problem zugrunde liegende Logik. Dabei vertiefen die

Schülerinnen und Schüler ihr Verständnis dafür, wie Konditionalitäten in Kombination mit anderen Elementen des Codes funktionieren.

Auch die Idee, dass es Grenzen für die Fähigkeiten eines Computers gibt, wird vorgestellt. Obwohl sich die Probleme in dieser Aktivität beide als "menschlich" herausstellen, ist es für die Schülerinnen und Schüler wertvoll zu wissen, dass es möglich ist, dass ein Problem eine Grenze der Fähigkeiten des Roboters darstellt, wenn sie mit komplexeren Projekten und Programmen arbeiten.

Tipps und Tricks

- Die Verwendung von Programmen, die Tastendrucke als Ereignisse innerhalb des Programms haben, kann verwirrend sein. Erinnern Sie die Schüler daran, dass sie, um ein Programm auszuführen, einmal auf die "Abspielen"-Taste (Dreieck) drücken müssen. Dieses Drücken startet NUR das Programm - es wird nicht als ein Knopfdruck innerhalb des Programms betrachtet. Mit anderen Worten, sobald das Programm läuft, sieht der Roboter den aktuellen Zustand als "Dreieckstaste gedrückt? - NEIN' AN.
- Die Problemlösung ist ein wichtiger Teil der Codierung. Diese Aktivität kann Schülern beim Üben des Problemlösens helfen, um ein "kaputtes" Programm zu reparieren, ohne dass sie sich so frustriert fühlen, wie wenn sie das Programm selbst geschrieben hätten.
- Zusammenarbeit kann das Finden von Lösungen beim Kodieren viel einfacher machen. Ziehen Sie in Betracht, Schüler in Paaren oder Gruppen an dieser Aktivität arbeiten zu lassen.
- Sie können diese Aktivität auf eine andere Art und Weise einrichten, um sie zu einer expliziteren Lektion über Zerlegung zu machen. Lassen Sie die Schüler in Gruppen oder als ganze Klasse arbeiten, wobei Sie sich zuerst die Kommentare des Programmierers ansehen sollten. Lassen Sie die Schülerinnen und Schüler die beiden getrennten Themen ohne Hilfe des Arbeitsblattes identifizieren und aufschlüsseln. Arbeiten Sie jede Ausgabe der Reihe nach durch und zerlegen Sie jedes Problem nach Bedarf in kleinere Einheiten. Lassen Sie die Schülerinnen und Schüler gemeinsam versuchen, jedes Niedrig-Teil zu lösen, und sehen Sie sich dann an, wie die anfänglichen Kommentare des Programmierers als Ganzes gelöst werden können.

Frage	Typ	Antwort/Beispielantwort	Hinweis
1	EA	<i>Der Roboter dreht sich langsam nach links.</i>	
2	SE	<i>Ja, denn das ist es, was der erste Block innerhalb der Schleife dem Roboter befiehlt.</i>	Wenn Schülerinnen und Schüler dieses Verhalten nicht erwartet haben, sollten Sie die Funktionsweise von Schleifen überprüfen.
3	EA	<i>Der Roboter fängt an, sich nach rechts zu drehen, weil die Bedingung des Blocks "Drehen nach links" erfüllt ist, und so geht der Roboter zum nächsten Block im Programm über, der "Drehen nach rechts" ist.</i>	Wenn Schüler nicht verstehen, warum es wieder nach links wandert, sollten Sie die Konditionalitäten überprüfen.
4	SE	<i>Der logische Fehler besteht darin, dass der Programmierer erwartete, dass die Schleife abbricht, sobald er den runden Knopf drückt, und der Roboter zur nächsten Codezeile übergeht (der Piepston). So funktionieren Schleifen jedoch nicht. Das Programm muss jede Codezeile innerhalb der Schleife abgeschlossen haben und dann an den Anfang der Schleife zurückkehren, um zu prüfen, ob die Schleifenbedingung erfüllt ist. Wenn die Bedingung erfüllt ist, macht das Programm weiter. Wenn der Programmierer den runden Knopf drückte, dann aber nicht den Dreiecksknopf, blieb das Programm in der ersten Codezeile innerhalb der Schleife stecken.</i>	Die Antworten der Schüler sollten den logischen Fehler korrekt identifizieren.

5	SE	<i>Liebes Zukunfts-Ich, zuerst einmal, geben Sie nicht auf. Kodieren erfordert Übung und Problemlösung. Hier sind noch einige andere Dinge, die Sie ausprobieren können: a) Gehen Sie Zeile für Zeile durch den Code und folgen Sie dabei dem, was jeder Block dem Roboter sagt. b) Schauen Sie sich an, wo das Problem liegt und testen Sie ein anderes Programm mit genau diesem Stück Code, um zu sehen, was passiert. c) Bitten Sie einen Freund, sich mein Programm anzusehen und zu sehen, ob er das Problem erkennen kann.</i>	Belastbarkeit ist der Schlüssel zur Programmierung! Vielleicht möchten Sie die Schüler daran erinnern, "nicht aufzugeben" in ihre Vorschlagsliste aufzunehmen!
---	----	---	--

Lösungsschlüssel

Aktivität U4-1.2 „Wenn“-Aussagen („If“ - Statements)

Umfang der Tätigkeit	Mittel
Empfehlung	
Benötigte Materialien	Arbeitsblatt U4-1.2, Tätigkeitsblatt U4-1

Überblick

Diese Offline-Aktivität stellt Schülern sowohl "Wenn"- als auch "„Wenn-Dann“-Bedingungen vor. Die Idee, dass Programme sich verzweigen können, wird erforscht, und Schüler üben das Konzept anhand einer Offline-"Schatzkarte" mit if-else-Anweisungen. Die Idee, dass Bedingungsanweisungen einem Computer erlauben, etwas zu entscheiden, ist bei der Verwendung von "Wenn"-Anweisungen offensichtlicher als bei der Verwendung von "Bis"-Bedingungen. Die Verzweigung wird besonders deutlich bei der Verwendung von if-else-Bedingungen. Die Übung des Konzepts der Offline-Verzweigung hilft den Schülern, sich mit der Idee besser vertraut zu machen und zu sehen, wie Verzweigungen in Kombination mit anderen rechnergestützten Denkprozessen wie Sequenz und Wiederholung funktionieren.

Tipps und Tricks

- Einige Schüler stellen vielleicht fest, dass das "Wenn-dann"-Muster, das in "Wenn"-Aussagen verwendet wird, dasselbe zu sein scheint wie die Kausalität. In gewisser Weise ist dies ein genauer Vergleich. Sobald der Code geschrieben ist, wird die 'DANN'-Anweisung ausgeführt, wenn die 'WENN'-Bedingung erfüllt ist. Im Code gibt es jedoch keine gesperrten 'Wenn-Dann'-Regeln. Vielleicht möchten Sie erklären, dass Schüler die 'Kausalität' durch die Bestimmung der Bedingungen (das WENN) und des bedingten Codes (das DANN) festlegen.

- Die Anweisungen zum Auffinden der Schätze sind alle in einem Format geschrieben, das als grober Pseudocode betrachtet werden könnte. Das Konzept des Pseudocodes wird in Aktivität U4-2.1 eingeführt. Sie haben die Wahl, das Konzept formell mit dieser Aktivität einzuführen oder die beiden Aktivitäten so zu verwenden, wie Sie sie jeweils verwenden.

Lösungsschlüssel

Frage	Typ	Antwort/Beispielantwort	Hinweis
1	SE	<i>Wenn mein Bruder in mein Zimmer geht, DANN verrate ich ihn.</i>	Jede Wenn-dann-Situation, die Sinn macht, ist akzeptabel.
2	EA	<i>E</i>	
3	EA	<i>B</i>	
4	EA	<i>D</i>	

Aktivität U4-1.3 „Wenn“-Aussagen und Sequenzen (If-Statements und Sequence)

Umfang der Tätigkeit	Niedrig
Empfehlung	
Benötigte Materialien	Basissatz, Arbeitsblatt U4-1.3

Überblick

Aufbauend auf dem Grundverständnis der Schülerinnen und Schüler von "Wenn"- und „Wenn-Dann“-Bedingungen untersucht diese Aktivität, wie "Wenn"-Aussagen in EdScratch-Programmen in Bezug auf die Sequenz funktionieren. Wie jeder Code folgen "Wenn"- und „Wenn-Dann“-Blöcke der grundlegenden Computerlogik, einschließlich der Ausführung in sequentieller Reihenfolge. Wenn das Programm auf 'if'-Anweisungen trifft, werden die Blöcke in schrittweiser sequentieller Reihenfolge aufgelöst. Da der Code innerhalb eines "Wenn"-Blocks jedoch davon abhängt, dass die Bedingung dieses Blocks erfüllt ist, kann das Programm den bedingten Code überspringen. Das Verständnis dieses Verhaltens ist für die Schülerinnen und Schüler von entscheidender Bedeutung, um Bedingungsbedingungen vollständig zu verstehen und die Logik der Bedingungen in ihrem rechnerischen Denken anwenden zu können.

Diese Aktivität untersucht auch „Wenn-Dann“-Blöcke in sequentiellen Programmen in EdScratch. Das Verständnis, dass nur der "Wenn"- oder "Dann"-Abschnitt des

Codes ausgeführt wird, ist für Schülerinnen und Schüler entscheidend, um Programme mit dieser Struktur schreiben und verfolgen zu können.

Tipps und Tricks

- Wenn Schüler Schwierigkeiten haben, den Geschehnissen in einem Programm mit "Wenn"- oder "Wenn-Dann"-Blöcken zu folgen, sollten Sie sie dazu ermutigen, Kommentare in ihre Programme aufzunehmen. Das Schreiben von Notizen in ihren eigenen Worten kann Schülern helfen, den bedingten Ablauf eines Programms zu verstehen.
- Erinnern Sie die Schüler daran, dass sie jederzeit ein Programm in EdScratch mit Edison ausführen können, um ihr Denken zu überprüfen und zu sehen, wie ein Programm funktioniert.

Frage	Typ	Antwort/Beispielantwort	Hinweis
1	EA	<i>Der runde Knopf muss gedrückt werden, damit die "Wenn"-Bedingung erfüllt ist.</i>	
2	SE	<i>Ich konnte den bedingten Code nicht zum Laufen bringen, selbst als ich den runden Knopf drückte. Als ich das Programm ausführte, piepte es und drehte sich dann um 60 Grad nach rechts. Es übersprang den Code innerhalb des 'if'-Blocks.</i>	Es ist unwahrscheinlich, dass Schüler in der Lage sind, den bedingten Code zum Laufen zu bringen, da nicht genug Zeit zwischen dem Drücken der Dreieckstaste zum Starten des Programms und der Überprüfung des runden Tastendrucks durch das Programm bleibt, um den Tastendruck abzuschließen. Wenn sie den Knopfdruck rechtzeitig beenden können, sollten ihre Antworten beachten, dass der gesamte Code nacheinander abläuft.
3	SE	<i>Ich denke, dass der bedingte Code übersprungen wird, wenn die Bedingung nicht erfüllt ist, und das Programm geht zum nächsten Codeelement nach dem 'if'-Block in sequentieller Reihenfolge über.</i>	In den Antworten der Schüler wird idealerweise vermerkt, dass der bedingte Code übersprungen wird, wenn die Bedingung nicht erfüllt ist.

4	EA	<i>Nein, der Roboter würde nicht piepsen. Dies liegt daran, dass sich der Piepston innerhalb des "Wenn"-Teils des Blocks befindet, der nur dann läuft, wenn der Roboter einen runden Tastendruck erkennt.</i>	
5	EA	<i>Es gibt 3 Aktionen, die der Roboter immer ausführt: 1) .5 Sekunden warten, 2) prüfen, ob ein runder Tastendruck erfolgt ist und 3) die linke LED einschalten.</i>	Wenn Schüler nur 1) und 3) identifizieren, möchten Sie vielleicht erklären, dass die Prüfung der Wenn-Aussage-Bedingung immer stattfindet, unabhängig vom Ergebnis.

□

Lösungsschlüssel

Aktivität U4-1.4 Wir untersuchen das Stapeln und Verschachteln von if-Anweisungen

Umfang der Tätigkeit	Groß
Empfehlung	
Benötigte Materialien	Basissatz, Arbeitsblatt U4-1.4

Überblick

Der effektive Einsatz von Konditionalen ermöglicht es Schülern, sinnvolle Programme zu erstellen und die leistungsstarken Sensoren von Edison zu nutzen. Das bedeutet, dass die Schüler ein klares Verständnis davon haben müssen, wie Konditionalitäten, insbesondere "Wenn"- und "Wenn-Dann"-Strukturen, in Programmen funktionieren, auch wenn diese Strukturen gestapelt oder verschachtelt sind.

In der Lage zu sein, dem Ablauf eines Programms zu folgen, das Konditional- und Schleifenkontrollstrukturen enthält, kann zunächst wie eine ziemlich imposante Aufgabe erscheinen. Diese Aktivität fordert Schüler dazu heraus, verschiedene Programme durchzuarbeiten und dabei dem Fluss des Codes zu folgen, um zu verstehen, wie die Programme funktionieren. Die Schüler setzen ihre rechnerischen

Denkfähigkeiten ein und zerlegen Programme, um zu verstehen, was in verschiedenen bedingten Situationen passieren wird. Dann sehen sie die Leistungsfähigkeit von in Schleifen verschachtelten Verzweigungs-Programmen, indem sie ein Klatsch-gesteuertes Fahrprogramm schreiben und testen.

Tipps und Tricks

- Die Verschachtelung von sensorbasierten Konditionals in Schleifen ist eine sehr gebräuchliche Art, Edisons Sensoren zu verwenden. Möglicherweise möchten Sie zusätzliche Zeit für diese Aktivität aufwenden, um sicherzustellen, dass die Schüler sich bei der Verwendung verschachtelter und gestapelter "Wenn"-Anweisungen wohl fühlen.
- Für das klatschgesteuerte Fahrprogramm: Die Roboter können Schwierigkeiten haben, Geräusche zu erkennen, wenn ein hoher Pegel von Hintergrundgeräuschen vorhanden ist. Wenn Schüler mit dem Finger in die Nähe des Schallsensors ihres Edison tippen, wird der gleiche Effekt wie beim Klatschen erzielt.
- Wenn Schüler ihrem klatschgesteuerten Fahrprogramm Kommentare hinzufügen, kann dies dazu beitragen, dass die Schüler verstehen, was innerhalb des Codes geschieht.
- Das klatschgesteuerte Fahrprogramm bei dieser Aktivität funktioniert auf dieselbe Weise wie das klatschgesteuerte Fahrprogramm, das durch das klatschgesteuerte Fahr-Strichcode-Programm aktiviert wird. Sie können sich dafür entscheiden, dass die Schüler diesen Strichcode erneut besuchen, um sich selbst davon zu überzeugen.

Lösungsschlüssel

Frage	Typ	Antwort/Beispielantwort	Hinweis
1	EA	<i>Wenn Sie niemals den runden Knopf drücken, wird die Bedingung niemals erfüllt, so dass der 'else'-Code bei jeder Wiederholung der Schleife ausgeführt wird. Der Roboter dreht sich um 60 Grad nach rechts, piepst dann und wiederholt diese Aktionen für immer.</i>	
2	EA	<i>Wenn Sie nie den runden Knopf drücken, wird die Bedingung nie erfüllt, so dass der 'wenn'-Code bei jeder Wiederholung der Schleife übersprungen wird. Der Roboter wird bei jeder Wiederholung der Schleife für immer piepsen.</i>	

3	EA	<i>Um den Roboter rückwärts fahren zu lassen, müssen Sie zuerst den runden Knopf drücken, während das Programm läuft. Der Roboter wartet .5 Sekunden und prüft dann, ob die Dreieckstaste gedrückt wurde. Da dies nicht der Fall ist, wird er den 'else'-Code ausführen und rückwärts fahren.</i>	
4	SE	<i>Ich denke, dass das Programm den 'forever'-Block verwendet, so dass das Programm Schleife für Schleife so lange läuft, wie Sie wollen. Wenn Sie den forever-Block nicht hätten, würde das Programm dem Roboter sagen, er solle auf einen Beifall warten und dann nach einem zweiten Beifall suchen, aber er würde dies nur einmal tun. Sobald er entweder einen oder zwei Beifälle erkennen und sich entsprechend bewegen würde, würde das Programm beendet werden.</i>	Schüler sollten erkennen, dass das Programm ohne die Dauersperre nur einmal laufen würde.

Activity U4-1.4a : Eine Riemenscheibe bauen

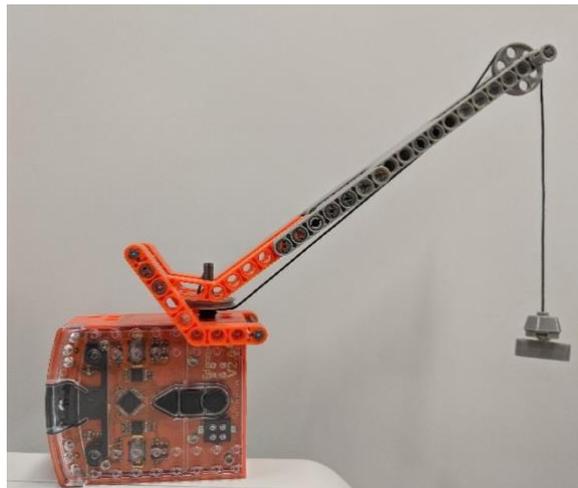
Umfang der Tätigkeit	Projekt
Empfehlung	Empfehlen Sie als Höhepunkt eine Aktivität nach Abschluss der Aktivitäten von Einheit 4, Lektion 1
Benötigte Materialien	Basissatz, Arbeitsblatt U4-1.4a, Makerspace/Zubehör, EdCreate-Sätze/LEGO-Steine

Überblick

Bei dieser offenen Ingenieurs- und Programmierherausforderung müssen Schüler ihr Verständnis der "Wenn"-Aussagen anwenden, um mit Edison ein funktionierendes Flaschenzugsystem zu schaffen.

Tipps und Tricks

- Diese Aktivität funktioniert gut in Paaren oder Kleingruppen.
- Vom Design her bietet diese Herausforderung minimale Anweisungen und nur wenige Hinweise. Dadurch sollen Kreativität und Experimentieren gefördert werden. Sie können sich dafür entscheiden, die Aktivität stärker zu strukturieren (z.B. den physischen Entwurfsprozess mit den Schülern durchzuarbeiten), wenn dies den Bedürfnissen Ihrer Schüler am besten entspricht.
- Diese Herausforderung kann als technische Designherausforderung durchgeführt werden. Sie können mehr über das Unterrichten von technischem Design unter <https://meetedison.com/teach-engineering-design-with-edcreate/> lesen.
- Es gibt keine "richtige" Antwort darauf, wie der Aufbau der Riemenscheibe und das Programm aussehen sollte, aber hier ist eine Beispiellösung, wie sie aussehen könnte:



- Physikalischer Aufbau: (angepasst an die [EdCrane EdBuild](#)):
- EdScratch-Riemenscheiben-Programm: <https://www.edscratchapp.com?share=wbJRrQbj>

Lösungsschlüssel

Frage	Typ	Antwort/Beispielantwort	Hinweis
1	SE	<p><i>Was das Programm betrifft, so waren in meiner ersten Version die "wenn"-Aussagen falsch verschachtelt. Beide befanden sich im "wenn"-Teil des "wenn"-Blocks, so dass die einzige Möglichkeit, die Rolle rückwärts zu bewegen, darin bestand, den runden Knopf und DANN den Dreiecksknopf zu drücken. Eine Klassenkameradin sah sich mein Programm an, und sie entdeckte diesen Fehler. Als ich den Dreiecksknopf "Wenn"-Block in den "Dann"-Abschnitt des „Wenn-Dann“-Blocks verschoben hatte, funktionierte es!</i></p> <p><i>Der beste Teil dieses Projekts war die Zusammenarbeit mit anderen Leuten. Wenn ich feststeckte und frustriert war, konnte ich Hilfe von Klassenkameraden bekommen, und dann haben wir die Probleme gemeinsam gelöst! Das war ein Riesenspaß.</i></p>	<p>Jede Erfahrung von Schülern ist akzeptabel. Sie können sich auch dafür entscheiden, dass Schüler Fotos oder Diagramme ihrer Riemenscheibenentwürfe zur Verfügung stellen und ihre Programmlösungen einreichen.</p>

Lektion 2: Fühlen

Diese relativ umfangreiche Lektion erweitert das Verständnis der Schüler für Bedingungen, Schleifen, Sequenzen und Unterbrechungen durch die Verwendung von Edisons verschiedenen Sensoren in Programmen. Die Schüler erforschen weiter, wie Eingaben in Edison funktionieren, indem sie neue Blöcke in den Kategorien "Kontrolle", "Sensorik" und "Ereignisse" in EdScratch verwenden.

Zusätzlich zur Entdeckung, wie Edisons individuelle Sensoren funktionieren und programmiert werden können, nutzt diese Lektion Edisons Sensorfähigkeiten, um wichtige Informatikkonzepte und -fähigkeiten in praktischen Anwendungen zu erforschen. Diese Lektion führt in die computerwissenschaftliche Praxis des Pseudocodes ein, in die Idee der Algorithmen und in das wichtige, aber oft missverständliche Konzept, wie sich Algorithmen von Programmen unterscheiden. Die Schüler üben die Verwendung von Pseudocode und Algorithmen sowohl zur Planung von Programmen als auch zur Unterstützung bei der Fehlersuche in ihrer Arbeit. Durch das Experimentieren mit Edisons Sensoren und Eingabemöglichkeiten in Arbeitsprogrammen erweitern die Schüler ihr Verständnis von Roboteranwendungen und autonomen Roboterverhaltensweisen in realen Kontexten.

Diese Lektion umfasst insgesamt fünf Basisaktivitäten und 13 Erweiterungsaktivitäten:

- [U4-2.1 Pseudocode](#)
 - o [U4-2.1a Antwort finden](#)
- [U4-2.2 Wir untersuchen Edisons Linienvorfolger](#)
 - o [U4-2.2a Innerhalb einer Grenze fahren](#)
- [U4-2.3 Wir untersuchen Algorithmen](#)
 - o [U4-2.3a Herausforderung angenommen: Es gibt mehr als einen Weg, einer Linie zu folgen](#)
- [U4-2.4 Edisons Hinderniserkennung untersuchen](#)
 - o [U4-2.4a Schneller, schneller, zerschlagen?](#)
 - o [U4-2.4b Herausforderung angenommen: Wenn Linie, nach rechts gehen. Wenn Hindernis, nach links gehen](#)
 - o [U4-2.4c Wo liegt das Hindernis?](#)
 - o [U4-2.4d Herausforderung angenommen: 3D-Labyrinth](#)
- [U4-2.5 Den Nachrichtenaustausch mit Edison untersuchen](#)

- o [U4-2.5a Ferngesteuerte Flaggenmaschine](#)
- o [U4-2.5b Herausforderung angenommen: Bau und Steuerung des EdCrane](#)
- o [U4-2.5c Herausforderung angenommen: Wasserwerfer zur Brandbekämpfung](#)
- o [U4-2.5d Halbautomatischer Bagger](#)
- o [U4-2.5e Herausforderung angenommen: Entfernung von gefährlichem Material](#)
- o [U4-2.5f Herausforderung angenommen: Brieftauben](#)

Activity U4-2.1 Let's explore pseudocode

Umfang der Tätigkeit	Klein
Empfehlung	
Benötigte Materialien	Arbeitsblatt U4-2.1, Tätigkeitsblatt U4-2

Überblick

Diese Offline-Aktivität führt Pseudocode ein, ein hilfreiches Werkzeug zur Bewältigung der wichtigen Aufgabe der Programmplanung. Das effektive Planen von Programmen ist eine wichtige Fähigkeit, die Schüler entwickeln müssen, wenn sie rechnergestütztes Denken beherrschen. Um effektiv planen zu können, müssen die Schüler die gelernten Konzepte des rechnergestützten Denkens (einschließlich Sequenz, Wiederholung und Bedingungen) in einen logischen Ablauf umsetzen. Pseudocode ist eines der effizientesten Werkzeuge zur Planung von Computerprogrammen und wird von vielen professionellen Programmierern verwendet.

Die Entwicklung eines Komfortniveaus mit Pseudocode ermöglicht es Schülern, Programme und Algorithmen mit Zuversicht zu planen. Das Arbeiten nach einem Plan macht die Codierung schneller und die Fehlersuche und Problemlösung effektiver. Die Möglichkeit, Programmpläne in Pseudocode zu erstellen, ermöglicht es Schülern auch, sich auf die Logik zu konzentrieren, die ihr Programm benötigt, ohne sich in kleinen Details oder der Syntax zu verlieren. Dies ermöglicht den Schülern viel mehr Klarheit bei der Planung von Programmen mit Sensoren, Daten oder komplexen Kontrollstrukturen. Die Kompetenz im Umgang mit Pseudocode verbessert auch die Fähigkeit der Schüler, ihr rechnerisches Verständnis auf andere Programmiersprachen, einschließlich textbasierter Sprachen, anzuwenden.

Tipps und Tricks

- Der Pseudocode im Arbeitsblatt verwendet Einrückungen, um anzuzeigen, wann Aktionen innerhalb von Kontrollstrukturen liegen. Wenn der Pseudocode im Arbeitsblatt ganz links ausgerichtet ist, helfen die Fragen im Arbeitsblatt zu zeigen, wo eine Schleife oder "if"-Anweisung endet und der nächste Befehl nach dieser Kontrollstruktur beginnt.
- Erinnern Sie die Schüler daran, dass sich der eingerückte Pseudocode innerhalb einer Schleife oder einer "Wenn"-Anweisung befindet. In Fällen, in denen es verschachtelte Bedingungen oder Schleifen gibt, zeigt der Pseudocode diese "Verschachtelung" an, indem er um mehr als eine Ebene eingerückt wird.

- Die Anweisungen zum Auffinden der Schätze in Aktivität U4-1.2 sind in einem Format geschrieben, das als grober Pseudocode betrachtet werden könnte. Sie können die Aktivität U4-1.2 als ein weiteres Beispiel für Pseudocode verwenden oder die beiden Aktivitäten zusammen verwenden.

Lösungsschlüssel

Frage	Typ	Answer	Hinweis
1	EA	<i>Der Buchstabe E</i>	
2	EA	<i>Die Zahl 4</i>	
3	EA	<i>Die Zahl 4</i>	

Aktivität U4-2.1a : Die Antwort finden

Umfang der Tätigkeit	Niedrig
Empfehlung	Beenden Sie die Aktivität U4-2.1 vor dieser Aktivität
Benötigte Materialien	- Arbeitsblatt U4-2.1a, Tätigkeitsblatt U4-2 - Optional: Computer oder Tablets zum Betrachten von EdScratch

Überblick

Diese Aktivität ist für die Verwendung in Verbindung mit der Aktivität U4-2.1 konzipiert und fördert das Verständnis und die Kompetenz der Schüler im Umgang mit Pseudocode. Die Schüler üben das Schreiben von Pseudocode-Anweisungen, dann das Lesen und Befolgen von Pseudocode, der von einem Klassenkameraden geschrieben wurde.

Tipps und Tricks

- Während diese Aktivität erfordert, dass die Schülerinnen und Schüler mit mindestens einem Partner zusammenarbeiten, können Sie die Schülerinnen und Schüler auch in Kleingruppen arbeiten lassen, die in Teams aufgeteilt sind. Jedes Team kann zusammenarbeiten, um den Pseudocode zu planen, zu schreiben und zu testen, bevor es sich mit dem anderen Team austauscht.

Lösungsschlüssel

Frage	Typ	Antwort/Beispielantwort	Hinweis
1	SE	<i>Start auf der 8 Richtung Norden 3 Mal wiederholen <i>vorwärts bis Brief rechts 90 2 Mal wiederholen <i>Stürmer 2 wenn Tier <i>rechts 90 Grad sonst <i>links 90 Grad vorwärts bis Brief Landet auf der Antwort von: C</i></i></i></i></i>	Sie können sich dafür entscheiden, dass Schüler auch beschreiben, wo ihr Programm enden soll, um die Genauigkeit zu überprüfen.

Aktivität U4-2.2 Wir untersuchen Edisons Linienverfolger

Umfang der Tätigkeit	Mittel
Empfehlung	
Benötigte Materialien	- Basissatz, Arbeitsblatt U4-2.2, Tätigkeitsblatt U4-3 - Fakultativ: EdMats, Zubehör für die Herstellung Ihrer eigenen Linien

Überblick

Diese zweiteilige Aktivität gibt Schülern den Anstoß zur Erforschung der Verwendung von Edisons Line Tracker in EdScratch-Programmen. Die Schüler machen sich zunächst mit der Linienverfolgungssensortechnologie des Edison-Roboters vertraut und lernen die Grundlagen der Funktionsweise des Sensors kennen. Durch das Experimentieren mit der roten LED des Linienverfolgungsgeräts auf verschiedenfarbigen Oberflächen lernen die Schülerinnen und Schüler, wie der Roboter den Sensor verwenden kann, um zu erkennen, ob er auf einer reflektierenden oder nicht reflektierenden Oberfläche fährt.

Die Schüler üben dann die Verwendung des Linienverfolgungssensors in einem EdScratch-Programm, um den Roboter zum Fahren zu bewegen, bis er eine schwarze (d.h. nicht reflektierende) Linie erkennt.

Tipps und Tricks

- Wenn Sie mehr darüber erfahren möchten, wie der Edison-Linienverfolgungssensor funktioniert, sollten Sie eine Breakout-Sitzung in Betracht ziehen, um zu untersuchen, wie Lichtreflexion und -absorption auch bei farbigem Licht funktionieren.
- Sie können auch EdMats verwenden oder Ihren eigenen Testraum einrichten, um das Programm "Fahren bis zum Schwarz" auszuführen.
- Erinnern Sie die Schüler daran, die Gewohnheit zu entwickeln, das Fehlerfeld in EdScratch anzukreuzen. Dies kann besonders bei der Arbeit mit Sensoren hilfreich sein.

Lösungsschlüssel

Frage	Typ	Answer	Hinweis
1	EA	<i>Eine weiße Oberfläche reflektiert mehr Licht zurück nach Edison als eine schwarze Oberfläche. Ich denke, das liegt daran, dass der Fleck der roten LED auf einer weißen Oberfläche heller ist (im Vergleich zu schwarz) und das bedeutet, dass mehr Licht reflektiert (und nicht absorbiert) wird.</i>	Das Licht wird auf einer weißen Fläche heller erscheinen. Wenn Schüler schwarz notieren, sollten Sie dies mit ihnen besprechen.
2	EA	Farbe reflektierend oder nicht reflektierend? Rot reflektierend Blau nicht reflektierend Grün nicht reflektierend	

Aktivität U4-2.2a : Fahrt innerhalb einer Grenze

Umfang der Tätigkeit	Mittel
Empfehlung	Beenden Sie die Aktivität U4-2.2 vor dieser Aktivität
Benötigte Materialien	- Basissatz, Arbeitsblatt U4-2.2a, Tätigkeitsblatt U4-4 - Fakultativ: EdMats, Zubehör für die Herstellung eigener Linien/Bordüren

Überblick

Schüler stellen ihr Verständnis von Pseudocode und Edisons Line Tracker auf die Probe, um ein Programm zu erstellen, das Edison dazu bringt, innerhalb eines schwarzen Rahmens zu fahren. Bei der Planung ihres Programms wenden die Schüler zunächst algorithmische Überlegungen an, wie der Linienverfolgungssensor verwendet werden kann, um ein Programm zu erstellen, bei dem Edison nicht über eine schwarze Linie fährt. Die Schüler übersetzen dann ihren Plan in EdScratch und testen ihr Programm mit dem Roboter.

Tipps und Tricks

- Sie können Schüler auch EdMats benutzen lassen oder ihren eigenen Testplatz einrichten lassen, um ihr "Laufwerk innerhalb eines schwarzen Rahmens"-Programme laufen zu lassen.
- Das Strichcode-Programm 'bounce in borders' hat das Edison-Laufwerk innerhalb eines schwarzen Rahmens. Sie können sich dafür entscheiden, dass Schüler das Strichcode-Programm ausführen und das Verhalten des Roboters bei der Verwendung des Strichcode-Programms mit dem Verhalten vergleichen, das beim Ausführen ihres Programms im Stil von 'Bounce in borders' angezeigt wird.
- Ermutigen Sie die Schüler dazu, die Fehlerquelle, auf die sie stoßen, herauszufinden. Zwei häufige logische Fehler, auf die Schüler bei der Erstellung dieses Programms stoßen können, sind 1) sie haben vergessen, den Zeilenerkennungssensor einzuschalten, oder 2) sie haben vergessen, eine unbestimmte Schleife zu verwenden.
- Diese Aktivität kann im Anschluss an die Aktivität U4-2.3 „Lassen Sie uns Algorithmen untersuchen“ verwendet werden, in der wir betonen, dass Schüler einen Algorithmus erstellen können, um Edison dazu zu bringen, innerhalb jeder beliebigen Grenze zu fahren.

Lösungsschlüssel

Frage	Typ	Antwort/Beispielantwort	Hinweis
1	SE	<i>Online-Erkennung einschalten für immer vorwärts fahren bis nicht reflektierend rückwärts fahren drehen</i>	Die Antworten der Schüler sind im Vergleich zum endgültigen Programm möglicherweise nicht ganz korrekt - das ist in Ordnung.

Aktivität U4-2.3 Untersuchen wir Algorithmen

Umfang der Tätigkeit	Mittel
Empfehlung	
Benötigte Materialien	Basissatz, Arbeitsblatt U4-2.3, Tätigkeitsblatt U4-4, Zubehör für die Herstellung eigener Linien/Bordüren und/oder EdMats

Überblick

Das Konzept der Algorithmen wird in dieser Lektion zusammen mit dem wichtigen, aber oft missverstandenen Konzept, wie sich Algorithmen von Programmen unterscheiden, eingeführt. Die Schüler lernen, dass Algorithmen ein Werkzeug zur Lösung von Problemstellungen sind und die in Computerprogrammen verwendete Logik lenken können. Sie erforschen einen Algorithmus, der Edison dazu bringt, mit Hilfe des Linienverfolgungssensors des Roboters jeder schwarzen Linie zu folgen, und sehen, wie sich die Logik des Algorithmus in ein EdScratch-Programm übersetzt.

Zu lernen, dass Algorithmen Computerprogramme informieren können, dass aber nicht alle Programme Algorithmen sind, ist ein wichtiges Konzept, um Schüler in die Lage zu versetzen, das so genannte "algorithmische Denken" zu üben. Zu verstehen, dass einige Probleme ein spezifisches, dediziertes Programm benötigen, um sie zu lösen, während andere mit Hilfe eines Algorithmus gelöst werden können, ist eine wichtige Fähigkeit, die die Schüler entwickeln müssen.

Tipps und Tricks

- Damit das Linienverfolgungsprogramm funktioniert, muss der Unterschied zwischen den dunklen und hellen Oberflächen für den Roboter leicht verständlich sein, und es muss genügend Weißraum zwischen allen Kurven in der Linie vorhanden sein. Die Linien, die die Schüler für Edison erstellen, müssen dunkel (z.B. schwarz), etwa 1,5 cm (0,6 Zoll) breit und auf weißem Hintergrund sein. Alternativ können sie einen EdMat verwenden.
- Stellen Sie sicher, dass die Schüler damit beginnen, Edison mit dem Linienverfolgungssensor auf weißem Hintergrund zu platzieren. Der Roboter kann in der Nähe der schwarzen Linie beginnen, aber nicht auf der schwarzen Linie.
- Sie können sich dafür entscheiden, dass die Schülerinnen und Schüler das Strichcode-Programm "einer Linie folgen" ausführen, um denselben Algorithmus vom Arbeitsblatt aus bei der Arbeit in diesem Strichcode-Programm zu sehen.
- Die Aktivität U4-2.2a kann im Anschluss an diese Aktivität als Erweiterungsaktivität verwendet werden. Sie können den Schülern demonstrieren, dass sie einen Algorithmus erstellen können, um Edison dazu zu bringen, innerhalb eines beliebigen schwarzen Rahmens zu fahren.

Lösungsschlüssel

Frage	Typ	Antwort/Beispielantwort	Hinweis
1	EA	<i>Der Roboter "watschelt" am Rand der Linie entlang. Dies liegt daran, dass der Algorithmus besagt, dass der Roboter vorwärts nach links fahren soll, bis sich der Linienverfolgungssensor auf einer nicht reflektierenden (schwarzen) Oberfläche befindet. Das schiebt den Sensor entlang, bis er eine schwarze Linie nach links findet. Dann sagt die Logik, dass der Roboter vorwärts nach rechts fahren soll, bis sich der Linienverfolgungssensor auf einer reflektierenden (weißen) Oberfläche befindet. Das schiebt den Sensor von der schwarzen auf die weiße Fläche. Der Roboter wechselt dieses Verhalten ab, indem er sich am Rand der Linie hin und her bewegt. Dadurch "watschelt" der Roboter links und rechts, während er die Linie entlang fährt.</i>	Die Schüler sollen erkennen, dass das Watscheln des Roboters dadurch verursacht wird, dass die Programmlogik zwischen den beiden "bis"-Blöcken hin- und hergeht.

2	SE	<i>Edison konnte meiner Linie folgen, außer an einer Stelle, wo die Linie sehr stark gekrümmt war. Der Roboter verfehlte die Kurve immer wieder und fing den anderen Teil der Linie auf. Ich glaube, dies geschah, weil die beiden Teile der Linie sehr nahe beieinander lagen, so dass der Roboter einfach eine nicht reflektierende Stelle fand und anfang, ihr zu folgen, obwohl es nicht der Teil der Linie war, dem ich ihn folgen lassen wollte.</i>	
---	----	--	--

Aktivität U4-2.3a : Es gibt mehr als einen Weg, einer Linie zu folgen

Umfang der Tätigkeit	Groß
Empfehlung	Beenden Sie die Aktivität U4-2.3 vor dieser Aktivität
Benötigte Materialien	<ul style="list-style-type: none"> - Basissatz, Arbeitsblatt U4-2.3a, Tätigkeitsblatt U4-4 - Fakultativ: EdMats, Zubehör für die Herstellung eigener Linien/Bordüren

Überblick

Die Idee, dass es keinen "richtigen" Weg zur Programmierung gibt, wird in dieser Aktivität mit Leben erfüllt. Schüler schauen sich den Algorithmus an, um Edison mit Hilfe des Linienverfolgungssensors des Roboters dazu zu bringen, jeder schwarzen Linie zu folgen, und verwenden dessen Logik, um mehrere Programme zu erstellen. Diese halboffene Programmieraufgabe fordert die Schülerinnen und Schüler auf, über verschiedene Möglichkeiten nachzudenken, wie die gleiche zugrunde liegende Logik in einem EdScratch-Programm manifestiert werden kann. Die Einsicht, dass es viele Wege gibt, eine Lösung zu programmieren, ist eine wichtige Fähigkeit, die Schülerinnen und Schüler entwickeln können und die sie besser in die Lage versetzt, effektiv an Codierungsprojekten zusammenzuarbeiten. Das Nachdenken über mehrere Möglichkeiten, die Logik des Algorithmus anzuwenden, fordert die Schüler auch heraus, ihr Verständnis von Bedingungen, Unterbrechungen und Sensoreingaben in Arbeitsprogrammen anzuwenden.

Tipps und Tricks

- Ermutigen Sie die Schüler dazu, die Logik des Algorithmus durcharbeiten und verschiedene Blöcke in EdScratch zu betrachten. Dinge zu testen, die am Ende nicht funktionieren, ist in Ordnung! Erforschen Sie, warum ein Versuch nicht funktioniert und iterieren Sie, um ein Programm zu entwickeln, das funktioniert.
- Damit das Linienverfolgungsprogramm funktioniert, muss der Unterschied zwischen den dunklen und hellen Oberflächen für den Roboter leicht verständlich sein, und es muss genügend Weißraum zwischen allen Kurven in der Linie vorhanden sein. Die Linien, die die Schüler für Edison erstellen, müssen dunkel (z.B. schwarz), ungefähr 1,5 cm (0,6 Inch) breit und auf weißem Hintergrund sein. Alternativ können sie einen EdMat verwenden.
- Stellen Sie sicher, dass die Schüler damit beginnen, Edison mit dem Linienverfolgungssensor auf weißem Hintergrund zu platzieren. Der Roboter kann in der Nähe der schwarzen Linie beginnen, aber nicht auf der schwarzen Linie.
- Vielleicht möchten Sie die Schüler die Antworten miteinander vergleichen lassen, um alle möglichen Programmvarianten zu sehen.

Lösungsschlüssel

Frage	Typ	Antwort/Beispielantwort	Hinweis
1	RC	<i>Programm 1: Dieses Programm verwendet eine „Wenn-Dann“-Anweisung, um Edison dazu zu bringen, einer Zeile zu folgen</i> https://www.edscratchapp.com?share=7bx9x9DZ	Jedes Programm, das Edison dazu bringt, irgendeiner Linie zu folgen, ist akzeptabel.
2	RC	<i>Programm 2: Dieses Programm nutzt Ereignisse, um Edison dazu zu bringen, einer Linie zu folgen</i> https://www.edscratchapp.com?share=pDmX9Jbq	Jedes Programm, das Edison dazu bringt, irgendeiner Linie zu folgen, ist akzeptabel.

Aktivität U4-2.4 Untersuchen wir Edisons Hinderniserkennung

Umfang der Tätigkeit	Groß
Empfehlung	
Benötigte Materialien	Basissatz, Arbeitsblatt U4-2.4, Objekte zur Herstellung von Hindernissen

Überblick

Diese dreiteilige Aktivität bildet den Auftakt zur Erforschung der Verwendung der Hinderniserkennung mit Edison in EdScratch-Programmen durch Schüler. Die Schülerinnen und Schüler machen sich zunächst mit der Infrarotlichtsensortechnologie des Edison-Roboters vertraut und lernen die Grundlagen der Funktionsweise des Sensors kennen und wie er zur Hinderniserkennung eingesetzt werden kann. Es wird auch ein grundlegendes Programm zur Hinderniserkennung vorgestellt, um zu erklären, wie der Sensor zur Erkennung von Hindernissen in EdScratch-Programmen verwendet werden kann. Schließlich üben die Schüler die Verwendung der Hinderniserkennung, indem sie einen Algorithmus zur Erkennung und Vermeidung von Hindernissen erstellen und dann die Logik des Algorithmus in ein EdScratch-Programm übersetzen.

Tipps und Tricks

- Wenn einer der Edison-Roboter Hindernisse nicht erkennt oder nur dann auf Hindernisse reagiert, wenn er sich sehr nahe am Hindernis befindet, müssen Sie möglicherweise die Hinderniserkennung in diesem Roboter kalibrieren. Verwenden Sie den Strichcode und die Anweisungen in Anhang 2 dieses Handbuchs, um die Hinderniserkennung eines Roboters zu kalibrieren.
- Hindernisse müssen undurchsichtig, aber nicht zu dunkel (z.B. nicht schwarz) und mindestens so hoch wie Edison sein, damit der Roboter sie erkennen kann.
- Sie können sich dafür entscheiden, dass Schüler das Strichcode-Programm "Hindernisse vermeiden" ausführen, um die gleiche algorithmische Logik in diesem Programm arbeiten zu sehen.
- Sie können sich für eine Breakout-Sitzung zum elektromagnetischen Spektrum entscheiden, bei der IR und sichtbares Licht in Verbindung mit dieser Aktivität in das Spektrum fallen.

Lösungsschlüssel

Frage	Typ	Antwort/Beispielantwort	Hinweis
1	EA	Hinderniserkennungsstrahl einschalten Schleife für immer vorwärts fahren wenn irgendwo ein Hindernis entdeckt wird rückwärts fahren drehen	Schülerantworten sollten die grundlegende Logik der Antwort/Beispielantwort haben, können aber anders erklärt werden.
2	SE	<i>Als ich anfang, meinen Algorithmus zu schreiben, musste ich darüber nachdenken, welches Verhalten der Roboter haben müsste, um jedem Hindernis auszuweichen. Das hat mich anfangs verblüfft. Ich nahm meinen Roboter in die Hand und tat so, als würde ich ihn auf dem Schreibtisch auf verschiedene Hindernisse zufahren, wobei ich mich bewegte, um jedem einzelnen auszuweichen. Indem ich dies einige Male tat, wurde mir klar, dass der Roboter, wenn er rückwärts fährt und sich dann wendet, immer dem Hindernis ausweicht.</i>	Wenn Schülerinnen und Schüler Probleme, auf die sie stoßen, sowie die Maßnahmen, die sie zur Lösung der Probleme ergriffen haben, artikulieren, hilft dies, ihre Problemlösungsfähigkeiten explizit hervorzuheben.

Aktivität U4-2.4a : Schneller, schneller, Klasse?

Umfang der Tätigkeit	Niedrig
Empfehlung	Beenden Sie die Aktivität U4-2.4 vor dieser Aktivität
Benötigte Materialien	Basissatz, Arbeitsblatt U4-2.4a, Objekte zur Herstellung von Hindernissen

Überblick

Schüler üben den Einsatz der Hinderniserkennung in Arbeitsprogrammen mit ihren Edison-Robotern, während sie die Ideen von Hardware-Beschränkungen und Kompromissen bei der Programmierung erforschen. Die Hinderniserkennung eines Edison-Roboters beruht darauf, dass der Roboter in der Lage ist, auf reflektiertes IR-Licht zu reagieren, das von Objekten auf ihn zurückprallt. Wenn sich der Roboter mit hoher Geschwindigkeit bewegt, kann die Zeit, die benötigt wird, um das Licht zu reflektieren und zu detektieren, die Zeit, die der Roboter benötigt, um auf das Hindernis zu prallen, übersteigen. Durch das Testen verschiedener Geschwindigkeiten in einem Hinderniserkennungsprogramm können Schüler diesen Kompromiss zwischen Geschwindigkeit und Genauigkeit aus erster Hand untersuchen.

Tipps und Tricks

- Wenn einer der Edison-Roboter Hindernisse nicht erkennt oder nur dann auf Hindernisse reagiert, wenn sie auch bei niedrigen Geschwindigkeiten sehr nahe sind, müssen Sie möglicherweise die Hinderniserkennung dieses Roboters kalibrieren. Verwenden Sie den Strichcode und die Anweisungen in Anhang 2 dieses Leitfadens, um die Hinderniserkennung eines Roboters zu kalibrieren.
- Hindernisse müssen undurchsichtig, aber nicht zu dunkel (z.B. nicht schwarz) und mindestens so hoch wie Edison sein, damit der Roboter sie erkennen kann.

Lösungsschlüssel

Frage	Typ	Antwort/Beispielantwort	Hinweis
1	SE	<i>Edison kann das Hindernis nicht rechtzeitig erkennen und stürzt sich darauf.</i>	Die Schüler sollten feststellen, dass der Roboter nicht in der Lage ist, Hindernisse zu erkennen und darauf zu reagieren, wenn Edison sich sehr schnell bewegt.
1	SE	<i>Je schneller Edison fährt, desto weniger Zeit bleibt für den Hinderniserkennungsprozess des Roboters. Je weniger Zeit für diesen Prozess zur Verfügung steht, desto schlechter ist die Fähigkeit des Roboters, Hindernisse zu erkennen.</i>	Schüler sollten in ihrer Antwort die Grundüberlegung in der Antwort/Beispielantwort beachten.

Aktivität U4-2.4b : Wenn Linie, nach rechts gehen. Wenn Hindernis, nach links gehen

Umfang der Tätigkeit	Mittel
Empfehlung	- Beenden Sie die Aktivität U4-2.2 vor dieser Aktivität - Beenden Sie die Aktivität U4-2.4 vor dieser Aktivität
Benötigte Materialien	- Basissatz, Arbeitsblatt U4-2.4b, Aktivitätsblatt U4-5, Objekte zur Herstellung von Hindernissen - Optional: Zubehör für die Herstellung Ihrer eigenen sensorgesteuerten 'Labyrinth'.

Überblick

Diese Programmieraufgabe wurde entwickelt, um eingesetzt zu werden, nachdem die Schüler sowohl Edisons Hinderniserkennungsstrahl als auch den Linienverfolgungssensor erforscht haben, und bringt die Schüler dazu, rechnerisches Denken anzuwenden, um mit Hilfe der Sensoren ein "Gitterlabyrinth" zu lösen. Die Schüler erforschen die Verwendung mehrerer Sensoren in einem einzigen Programm. Diese Aufgabe erfordert auch, dass sie mehrere Berechnungselemente

kombinieren, einschließlich Sequenz, Bedingungen, Ein- und Ausgaben und Wiederholung.

Die Aktivität bietet den Schülern auch die Möglichkeit, ihre eigene physikalische Anordnung zu entwerfen, die nach der gleichen Grundidee zu lösen ist: "Wenn Linie, dann X. Wenn Hindernis, dann Y."

Tipps und Tricks

- Wenn einer der Edison-Roboter Hindernisse nicht erkennt oder nur dann auf Hindernisse reagiert, wenn sie auch bei niedrigen Geschwindigkeiten sehr nahe sind, müssen Sie möglicherweise die Hinderniserkennung dieses Roboters kalibrieren. Verwenden Sie den Strichcode und die Anweisungen in Anhang 2 dieses Leitfadens, um die Hinderniserkennung eines Roboters zu kalibrieren.
- Hindernisse müssen undurchsichtig, aber nicht zu dunkel (z.B. nicht schwarz) und mindestens so hoch wie Edison sein, damit der Roboter sie erkennen kann.
- Eine Beispiellösung unter Verwendung des Aktivitätsblattes ist unter <https://www.edscratchapp.com?share=8DQGkG0B> zu sehen.

Aktivität U4-2.4c : Wo liegt das Hindernis?

Umfang der Tätigkeit	Niedrig
Empfehlung	Beenden Sie die Aktivität U4-2.4 vor dieser Aktivität
Benötigte Materialien	- Basissatz, Arbeitsblatt U4-2.4c, Objekte zur Herstellung von Hindernissen

Überblick

Diese Aktivität stärkt das Verständnis der Schülerinnen und Schüler dafür, wie der Infrarotsensor von Edison zur Erkennung von Objekten eingesetzt werden kann, und untersucht die Fähigkeit von Edison, die Position eines Hindernisses relativ zum Roboter zu erkennen. Mit der Anforderung, Unterbrechungen aus der Kategorie "Ereignisse" von Blöcken in EdScratch zu verwenden, nutzen die Schülerinnen und Schüler ihre Kreativität, um drei verschiedene Unterroutinen für den Roboter zu erstellen, die je nach der Position des erkannten Hindernisses unterschiedliche Reaktionen ausgeben.

Tipps und Tricks

- Wenn einer der Edison-Roboter Hindernisse nicht erkennt oder nur dann auf Hindernisse reagiert, wenn sie sehr nahe sind, müssen Sie möglicherweise die Hinderniserkennung dieses Roboters kalibrieren. Verwenden Sie den Strichcode und die Anweisungen in Anhang 2 dieses Handbuchs, um die Hinderniserkennung eines Roboters zu kalibrieren.
- Eines der besten Hindernisse für diese Aktivität sind eigentlich die eigenen Hände der Schülerinnen und Schüler! Lassen Sie die Schülerinnen und Schüler ihren Edison von sich weg sehen, dann das Programm testen, indem sie eine Hand vor dem Edison nach rechts legen und dann auf der linken Seite wiederholen. Schließlich sollen sie ihre Hand direkt vor den Roboter halten.
- Eine Beispiellösung für diese Aktivität ist zu sehen unter [„https://www.edscratchapp.com?share=Eb2X7wYq](https://www.edscratchapp.com?share=Eb2X7wYq)

Aktivität U4-2.4d : 3D-Labyrinth

Umfang der Tätigkeit	Projekt
Empfehlung	Beenden Sie die Aktivität U4-2.4 vor dieser Aktivität
Benötigte Materialien	Basissatz, Arbeitsblatt U4-2.4d, Zubehör für den Aufbau des 3D-Labyrinths

Überblick

Indem sie ihre eigenen dreidimensionalen Labyrinth erstellen und dann lösen, wenden die Schüler computergestütztes Denken und ihr Verständnis von Edisons infrarotgesteuerter Hinderniserkennung auf dieses "reale" Projekt an und bringen Edison dazu, wie ein fahrerloses Auto zu funktionieren. Genauso wie die Technologie des fahrerlosen Autos noch perfektioniert werden muss, ist es nicht ohne Fallstricke, Edison dazu zu bringen, mit Hilfe der Hinderniserkennung autonom durch ein 3D-Labyrinth zu fahren. Dieses Projekt ist eine große Herausforderung für Schüler, um Problemlösung, Zerlegung und Zusammenarbeit zu üben.

Tipps und Tricks

- Ein großartiges Beispiel für ein 3D-Labyrinth, das mit Hilfe der Hinderniserkennung leicht gelöst werden kann, ist in dem Video Meet Edison

- Autonomous Robotics Maze Challenge #1 zu sehen, das unter <https://www.youtube.com/watch?v=caVNQYKr-4> verfügbar ist.

- Erinnern Sie die Schüler an den Kompromiss zwischen Geschwindigkeit und Genauigkeit bei der Hinderniserkennung: Je schneller Edison fährt, desto weniger Zeit bleibt für die Hinderniserkennung des Roboters. Je weniger Zeit für diesen Prozess zur Verfügung steht, desto schlechter ist die Fähigkeit des Roboters, Hindernisse zu erkennen.
- Die Hinderniserkennung des Edison verwendet Infrarotlicht, das bei der Bewegung des Edison weit davon entfernt sein kann, perfekt zu sein. Wenn sich der Roboter bewegt, ändert sich seine Position relativ zu Objekten. „Altes“ Infrarotlicht, das von einer Wand abprallt, kann von einer neuen Position aus erkannt werden, wodurch der Roboter in einer Weise reagiert, die der Programmierer vielleicht nicht erwartet oder will. Das Experimentieren mit dem Einsatz von ortsabhängigen Hinderniserkennungsereignissen auf unterschiedliche Weise ermöglicht es den Schülern, verschiedene Optionen zu entwickeln, um ihre Roboter autonom durch ihr Labyrinth zu bringen.

Aktivität U4-2.5 Wir untersuchen die Nachrichtenübermittlung mit Edison

Umfang der Tätigkeit	Mittel
Empfehlung	Es wird empfohlen, Aktivität U4-2.4 vor dieser Aktivität abzuschließen.
Benötigte Materialien	Basissatz, Arbeitsblatt U4-2.5

Überblick

Diese Aktivität führt Schüler in die Verwendung des Infrarotsensors des Edison-Roboters zum Senden und Empfangen von IR-Nachrichten ein. Der IR-Empfänger des Roboters kann auf mehrere Arten verwendet werden: zur Erkennung von IR-Licht, das von Objekten als Teil des Hinderniserkennungsstrahls von Edison zum Roboter zurückreflektiert wird, zur Erkennung von IR-Nachrichten von anderen Edison-Robotern und zur Erkennung von IR-Nachrichten von gepaarten TV/DVD-Fernbedienungen. Die Schüler lernen die Grundlagen der Funktionsweise von IR-Nachrichten in dieser Aktivität und üben das Senden und Empfangen von Nachrichten zwischen mehreren Edison-Robotern.

Tipps und Tricks

- Für diese Aktivität sind Sätze von mindestens zwei Robotern erforderlich, daher sollten Schüler in Paaren oder Gruppen arbeiten. Die Aktivität kann auch als gesamte Klassenaktivität durchgeführt werden, wobei ein Roboter eine Nachricht an alle anderen Roboter sendet, um sie zu erkennen.
- Edisons Nachrichtenübermittlung verwendet Infrarot, das eine begrenzte Reichweite hat, ähnlich der einer TV-Fernbedienung. Wenn die Schülerinnen und Schüler Schwierigkeiten haben, einen "empfangenden" Roboter dazu zu bringen, die Nachricht zu erkennen, lassen Sie die Schülerinnen und Schüler die "empfangenden" Roboter näher an den "sendenden" Roboter heranrücken.
- Wenn mehrere Gruppen diese Aktivität nahe beieinander ausführen, können die Schülerinnen und Schüler auf „Übersprechen“ mit Robotern stoßen, die auf die von anderen Gruppen gesendeten Nachrichten reagieren.
- Um mehr über TV/DVD-Fernbedienungen und Edison zu erfahren, können Sie die Aktivität U1-1.2c : TV-Fernbedienungs-Barcodes und/oder die Aktivität U4-2.5a : *Ferngesteuerte Flaggenmaschine* verwenden.

Lösungsschlüssel

Frage	Typ	Antwort	Hinweis
1	EA	<i>Der Block "IR-Nachricht senden" gehört zur Kategorie LEDs in EdScratch, weil es sich um eine Ausgabe handelt, die Edisons Infrarot-LEDs verwendet.</i>	

Aktivität U4-2.5a : Ferngesteuerte Flaggenmaschine

Umfang der Tätigkeit	Projekt
Empfehlung	- Beenden Sie die Aktivität U4-2.5 vor dieser Aktivität - Bei den Aktivitäten U4-2.5a, U4-2.5b, U4-2.5c, U4-2.5d und U4-2.5e handelt es sich allesamt um Robotik-Bauprojekte, bei denen IR-Meldungen verwendet werden. Erwägen Sie die Verwendung von mindestens einer dieser Nachrichten.

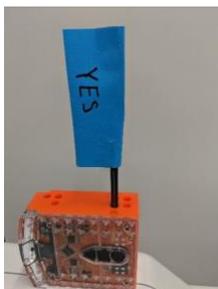
Benötigte Materialien	Basissatz, Arbeitsblatt U4-2.5a, Tätigkeitsblatt U4-6, TV/DVD-Fernbedienungen, Makerspace-Zubehör zum Erstellen der Flaggen, EdCreate-Sets und/oder anderes Zubehör zum Anbringen der Flaggen
------------------------------	---

Überblick

Die Schüler lernen in dieser Aktivität, wie man Edisons programmierbare TV-Fernbedienungs-codes verwendet, und wenden dieses Wissen dann in einem Roboterbauprojekt an, um eine ferngesteuerte Flaggenmaschine zu erstellen. Dieses halboffene Projekt fordert die Schüler dazu heraus, eine physikalische Computerlösung mit realer Anwendung zu erstellen und dabei ihr Verständnis von Codierungskonzepten, einschließlich Konditionierung und Abtastung, anzuwenden.

Tipps und Tricks

- Schülerinnen und Schüler sind herzlich eingeladen, eine Flag-Maschine auf eine andere Weise als in der Projektbeschreibung dieser Aktivität zu bauen oder eine eigene ferngesteuerte Code-Erkennung zu erstellen.
- Möglicherweise möchten Sie die Schüler mit der Verwendung von TV/DVD-Fernbedienungen und Edison mit der Aktivität U1-1.2c : TV-Fernbedienungs-Strichcodes vor dieser Aktivität vertraut machen.
- Eine Variation dieser Aktivität unter Verwendung von IR-Meldungen mit Variablen ist in Einheit 5, Aktivität U5-1.4b : Edison-gesteuerte Flaggenmaschine, enthalten. Sie haben die Wahl, auf diese beiden Aktivitäten bei der Verwendung einer der beiden Aktivitäten hinzuweisen oder sie als zwei Programmierherausforderungen für den Entwurf einer einzigen Konstruktion auszuführen.
- Es gibt keine "richtige" Antwort, aber hier ist, wie eine Beispiellösung aussehen könnte:
 - Entwurf:
 - Programm: <https://www.edscratchapp.com?share=RDaeaaDV>



Aktivität U4-2.5b : Bau und Steuerung des EdCrane

Umfang der Tätigkeit	Projekt
Empfehlung	- Beenden Sie die Aktivität U4-2.5 vor dieser Aktivität - Bei den Aktivitäten U4-2.5a, U4-2.5b, U4-2.5c, U4-2.5d und U4-2.5e handelt es sich allesamt um Robotik-Bauprojekte, bei denen IR-Meldungen verwendet werden. Erwägen Sie die Verwendung von mindestens einer dieser Nachrichten.
Benötigte Materialien	Basissatz, Arbeitsblatt U4-2.5b, Aktivitätsblatt U4-6, TV/DVD-Fernbedienungen, 1 EdCreate Bausatz pro EdCrane, EdCrane Bauanleitung (erhältlich unter https://meetiedison.com/content/EdCreate/EdBuild-EdCrane-instructions.pdf)

Überblick

Die Verwendung von Edison zur Erstellung programmierbarer Roboterbauten ist eines der aufregendsten Dinge, die Schüler mit den Robotern tun können. Die EdBuild-Projekte, bei denen die voreingestellten Anweisungen und EdCreate Bausätze verwendet werden, sind eine ausgezeichnete Möglichkeit, Schüler mit interaktiver Technik vertraut zu machen.

Wie andere EdCreate EdBuilds reagiert auch der EdCrane auf eine TV- oder DVD-Fernbedienung. Der EdCrane verwendet jedoch die programmierbaren Codes der TV-Fernbedienung und nicht die voreingestellten Aktions-Barcodes. Der EdCrane verwendet Fernbedienungsbefehle in Kombination mit einem von EdScratch heruntergeladenem Programm, das dem EdCrane mitteilt, welche Aktion ausgeführt werden soll, wenn ein bestimmter Fernbedienungscode erkannt wird.

Tipps und Tricks

- Dieses Projekt eignet sich ideal für Paar- oder Gruppenarbeit.
- Schülerinnen und Schüler finden es vielleicht am einfachsten, zunächst alle EdCreate-Stücke auf ihrer Arbeitsfläche auszulegen und die Teile in Gruppen desselben Stücktyps und derselben Farbe zu organisieren. Dies kann den Schülern dabei helfen, die richtigen Teile zu identifizieren und zu verwenden, während sie sich durch den Bau arbeiten.
- Es wird dringend empfohlen, die Magnethaken vor dem Bau unter Aufsicht von Erwachsenen aufzustellen, um sicherzustellen, dass diese korrekt aufgestellt werden. Sobald der Haken zum ersten Mal aufgestellt ist, kann er als Einzelteil immer wieder im EdCrane-Aufbau verwendet werden.
- Die Spule des Magnethakens kann sich sowohl im als auch gegen den Uhrzeigersinn drehen. Wann immer die Schnur des Hakens vollständig aus

der Spule herausgezogen ist (d.h. die Schnur ganz losgelassen wird), wird die Richtung, in die sich der Haken zu wickeln beginnt, zur Richtung "Haken aufheben". Die entgegengesetzte Richtung ist die Richtung des "unteren Hakens".

- Erinnern Sie die Schüler daran, dass sie die Abspielen-Taste (Dreieck) am Edison-Roboter drücken müssen, um ihr heruntergeladenes Programm zu starten und das Standardverhalten des Strichcodes außer Kraft zu setzen.
- Wenn mehrere Schülerinnen und Schüler Programme mit den Fernbedienungscodes in unmittelbarer physischer Nähe zueinander ausführen, kann es zu „Übersprechen“ kommen, wenn ein Roboter den Fernbedienungscodes einer anderen Gruppe empfängt und darauf reagiert. Versuchen Sie, die Schülerinnen und Schüler auf Abstand voneinander zu halten, um das Problem zu minimieren.
- Die Schülerinnen und Schüler können ihre EdScratch-Programme vor oder nach dem Bau des Krans in den EdCrane herunterladen, aber sie müssen den Roboter vor dem Bau mit einer Fernsteuerung unter Verwendung der Strichcodes koppeln.
- Die Bauanleitung für den EdCrane enthält auch Programmieranweisungen, einschließlich eines Links zu einem Freigabecode. Sie können sich dafür entscheiden, dass Schüler diesen Code verwenden, das Programm entsprechend ihrem Plan modifizieren und die Eingaben nach Bedarf anpassen. Alternativ können Sie den Schülern einen modifizierten Satz der EdCrane-Bauanleitung ohne den Freigabecode geben und die Schüler versuchen lassen, die Lösung selbst zu programmieren. Das Arbeitsblatt geht davon aus, dass Schüler NICHT über den Freigabecode verfügen. Der Freigabecode kann unter <https://www.edscratchapp.com?share=5DMQ3XDw> eingesehen werden.

Lösungsschlüssel

Frage	typ	Antwort/Beispielantwort			Hinweis
1	SE	EdCrane action	Remote code	Remote control button	
		Spin the magnetic spool clockwise	1	Volume up	
		Spin the magnetic spool counter-clockwise	2	Volume down	
		Spin the crane clockwise	3	Keypad right	
		Spin the crane counter-clockwise	4	Keypad left	

Activity U4-2.5c : Firefighting water cannon

Umfang der Tätigkeit	Projekt
Empfehlung	- Beenden Sie die Aktivität U4-2.5 vor dieser Aktivität - Bei den Aktivitäten U4-2.5a, U4-2.5b, U4-2.5c, U4-2.5d und U4-2.5e handelt es sich allesamt um Robotik-Bauprojekte, bei denen IR-Meldungen verwendet werden. Erwägen Sie die Verwendung von mindestens einem.
Benötigte Materialien	Basissatz, Arbeitsblatt U4-2.5c, 1 EdCreate-Satz pro EdTank, 2 Edison-Roboter pro EdTank, EdTank-Bauanleitungssatz (verfügbar unter https://meet Edison.com/content/EdCreate/EdBuild-EdTank-instructions.pdf), Zubehör zur Markierung der Situationseinrichtung

Überblick

Diese Programmierherausforderung verlangt von den Schülern, ihr Verständnis von Programmierkonzepten, einschließlich der Bedingung und Abtastung, auf eine Anwendung in der realen Welt anzuwenden. Die Projekte der EdBuild-Programmierherausforderung verwenden die voreingestellten Anweisungen und EdCreate-Kits, um das EdBuild zu erstellen, das für das Szenario benötigt wird. In diesem Szenario wird der EdTank verwendet. Durch die Verwendung des EdTank als Basis für ihre Feuerlöschroboter brauchen die Schülerinnen und Schüler die technische Komponente nicht zu entwerfen und können sich auf die Programmierkomponente ihrer Lösungen konzentrieren.

Die Schüler müssen zwei Programme entwerfen, die miteinander zusammenarbeiten, um die beiden Roboter im EdTank zu steuern und das Brandbekämpfungsszenario zu lösen.

Tipps und Tricks

- Dieses Projekt eignet sich ideal für Paar- oder Gruppenarbeit.
- Die Bauherausforderung in dieser Aktivität und in der Aktivität U1-1.2e: Den EdTank bauen und kontrollieren ist die gleiche.
- Die Schülerinnen und Schüler finden es vielleicht am einfachsten, zunächst alle EdCreate-Stücke auf ihrer Arbeitsfläche auszulegen und die Teile in Gruppen desselben Stücktyps und derselben Farbe zu organisieren. Dies kann den Schülern helfen, die richtigen Teile zu identifizieren und zu verwenden, während sie sich durch den Bau arbeiten.

- Sie müssen die Kanone zurücksetzen und jedes Mal, wenn Sie die Kanone abfeuern, ein Gummiband neu laden. Dies sollte manuell erfolgen, um sicherzustellen, dass das neue Band richtig geladen ist und der Schlagbolzen vollständig in die Ausgangsposition zurückgedrückt wird.
- Die besten Ergebnisse erzielen Sie mit den orangefarbenen Bändern, die mit Ihrem EdCreate Kit in der Kanone mitgeliefert werden.
- Wie viele echte Panzer ist der EdTank so konstruiert, dass er sich nur langsam nach links oder rechts drehen kann. Wenn die EdCreate-Ketten brandneu sind, können sie besonders griffig sein, wodurch der EdTank noch langsamer dreht. Sie können die Griffigkeit verringern, indem Sie die Raupen entfernen und sie leicht mit Talkumpuder einstauben. Achten Sie darauf, überschüssigen Puder von den Raupen abzuklopfen, bevor Sie sie wieder auf den EdTank legen.
- Weitere Informationen über dieses Programmierungsprojekt finden Sie im EdCreate Lehrhandbuch unter <https://meetedison.com/content/EdCreate/EdCreate-teachers-guide.pdf>.
- Es gibt keine "richtige" Antwort, und die Lösungen werden je nach den von Schülern geschaffenen Einrichtungen variieren. Eine Beispiellösung sieht jedoch wie folgt aus: <https://www.edscratchapp.com?share=80vBNQ0d>

Activity U4-2.5d : Semi-automated digger

Umfang der Tätigkeit	Projekt
Empfehlung	<ul style="list-style-type: none"> - Beenden Sie die Aktivität U4-2.5 vor dieser Aktivität - Verwenden Sie Aktivität U4-2.5a zur Einführung der Verwendung von Fernsteuerungscodes in Programmen vor dieser Aktivität - Bei den Aktivitäten U4-2.5a, U4-2.5b, U4-2.5c, U4-2.5d und U4-2.5e handelt es sich allesamt um Robotik-Bauprojekte, bei denen IR-Meldungen verwendet werden. Erwägen Sie die Verwendung von mindestens einem.

Benötigte Materialien	Basissatz, Arbeitsblatt U4-2.5d, Aktivitätsblatt U4-6, TV/DVD-Fernbedienungen, 1 EdCreate Bausatz pro EdDigger, 2 Edison-Roboter pro EdDigger, EdDigger Bauanleitungssatz (erhältlich unter https://meet Edison.com/content/EdCreate/EdBuild-EdDigger-instructions.pdf), Zubehör zur Erstellung der Situationsaufstellung
------------------------------	---

Überblick

Diese Programmierherausforderung verlangt von den Schülern, ihr Verständnis von Codierungskonzepten, einschließlich der Konditionierung und Abtastung, auf eine Anwendung in der realen Welt anzuwenden. Die Projekte der EdBuild-Programmierherausforderung verwenden die voreingestellten Anweisungen und EdCreate-Kits, um das EdBuild zu erstellen, das für das Szenario benötigt wird. Dieses Szenario verwendet den EdDigger. Durch die Verwendung des EdDigger als Basis für ihre halbautomatischen Baggerroboter brauchen die Schüler die Konstruktionskomponente nicht zu entwerfen und können sich auf die Programmierkomponente ihrer Lösungen konzentrieren.

Die Schüler müssen zwei Programme entwerfen, die in Verbindung miteinander arbeiten, um die beiden Roboter im EdDigger zu steuern und das Szenario zu lösen.

Tipps und Tricks

- Dieses Projekt eignet sich ideal für Paar- oder Gruppenarbeit.
- Die bauliche Herausforderung in dieser Aktivität und in der Aktivität U1-1.2f : Den EdDigger bauen und kontrollieren ist die gleiche.
- Die Schülerinnen und Schüler finden es vielleicht am einfachsten, zunächst alle EdCreate-Stücke auf ihrer Arbeitsfläche auszulegen und die Teile in Gruppen desselben Stücktyps und derselben Farbe zu organisieren. Dies kann den Schülern helfen, die richtigen Teile zu identifizieren und zu verwenden, während sie sich durch den Bau arbeiten.
- Bei dieser Konstruktion verbindet sich der obere Roboter mit dem unteren Roboter in der dritten Bolzenreihe von der Vorderseite des unteren Roboters aus. Auf diese Weise überragt der obere Roboter die Rückseite des unteren Roboters um etwa 2 cm.
- Weitere Informationen zu diesem Programmierprojekt finden Sie in der EdCreate-Lehranleitung unter <https://meet Edison.com/content/EdCreate/EdCreate-teachers-guide.pdf>.

- Es gibt keine "richtige" Antwort, und die Lösungen werden je nach den von Schülern geschaffenen Einrichtungen variieren. Eine Beispiellösung sieht jedoch wie folgt aus <https://www.edscratchapp.com?share=5DMe54Dw>

Activity U4-2.5e : Entfernung von gefährlichem Material

Umfang der Tätigkeit	Projekt
Empfehlung	- Beenden Sie die Aktivität U4-2.5 vor dieser Aktivität - Bei den Aktivitäten U4-2.5a, U4-2.5b, U4-2.5c, U4-2.5d und U4-2.5e handelt es sich allesamt um Robotik-Bauprojekte, bei denen IR-Meldungen verwendet werden. Erwägen Sie die Verwendung von mindestens einem.
Benötigte Materialien	Basissatz, Arbeitsblatt U4-2.5e, 1 Bausatz EdCreate pro EdRoboClaw, 2 Edison-Roboter pro EdRoboClaw, Bauanleitung für EdRoboClaw (erhältlich unter https://meetiedison.com/content/EdCreate/EdBuild-EdRoboClaw-instructions.pdf), Material zur Erstellung des Situationsaufbaus

Überblick

Diese Programmierherausforderung verlangt von den Schülern, ihr Verständnis von Codierungskonzepten, einschließlich der Konditionierung und Abtastung auf eine Anwendung in der realen Welt anzuwenden. Die Projekte der EdBuild-Programmierherausforderung verwenden die voreingestellten Anweisungen und EdCreate-Kits, um das EdBuild zu erstellen, das für das Szenario benötigt wird. Dieses Szenario verwendet das EdRoboClaw. Durch die Verwendung von EdRoboClaw als Basis für ihre Gefahrstoffentfernungsroboter brauchen die Schüler die technische Komponente nicht zu entwerfen und können sich auf die Programmierkomponente ihrer Lösungen konzentrieren.

Die Schüler müssen zwei Programme entwerfen, die in Verbindung miteinander arbeiten, um die beiden Roboter im EdRoboClaw zu steuern und das Szenario zu lösen.

Tipps und Tricks

- Dieses Projekt eignet sich ideal für Paar- oder Gruppenarbeit.
- Die bauliche Herausforderung bei dieser Aktivität und bei der Aktivität U1-1.2g: Bau und Kontrolle des EdRoboClaw ist die gleiche.

- Die Schülerinnen und Schüler finden es vielleicht am einfachsten, zunächst alle EdCreate-Stücke auf ihrer Arbeitsfläche auszulegen und die Teile in Gruppen desselben Stücktyps und derselben Farbe zu organisieren. Dies kann den Schülern helfen, die richtigen Teile zu identifizieren und zu verwenden, während sie sich durch den Bau arbeiten.
- Bei dieser Konstruktion verbindet sich der obere Roboter mit dem unteren Roboter auf der zweiten Bolzenreihe von der Vorderseite des unteren Roboters aus. Dadurch überragt der obere Roboter die Rückseite des unteren Roboters um etwa 1 cm.
- Die Klaue im Gelenkarm des EdRoboClaw besteht aus 3 "Fingern" - zwei parallelen Fingern, die stationär sind (aus grauen Balken), und dem vordersten Finger, der sich bewegt. Die Zahnradreihe im Gelenkarm steuert diesen vorderen Finger, einschließlich seiner Positionierung relativ zu den stationären Fingern. Die Ausrichtung der beiden vordersten Zahnräder kann sich darauf auswirken, wie der bewegliche Finger relativ zu den stationären Fingern sitzt, wenn die Klaue vollständig geöffnet ist. Wenn die Klaue vollständig geöffnet ist und der EdRoboClaw auf einer flachen Oberfläche (z.B. einem Tisch oder Schreibtisch) sitzt, sollte der bewegliche Finger hoch genug sein, dass einer der grauen EdCreate Balken zwischen Finger und Tisch darunter gleiten kann. Wenn der bewegliche Finger nicht so hoch ist, versuchen Sie, die Vorderseite des Arms vorsichtig zu trennen und das vorderste Zahnrad unabhängig vom nächsten Zahnrad um ein oder zwei Zähne im Uhrzeigersinn zu drehen. Sie werden sehen können, wie sich der vordere Finger dadurch bewegt. Verbinden Sie die Zahnräder und den Arm wieder miteinander.
- Der EdRoboClaw kann Objekte mit flachen Oberflächen aufnehmen. Es kann vorkommen, dass Schüler feststellen, dass sie runde Gegenstände, wie z.B. einen Stift, nicht so gut aufnehmen und tragen können. Versuchen Sie es mit dem 7-Loch langen grauen Balken aus dem EdCreate Kit.
- Am einfachsten ist es, den Roboter mit der Klaue zu starten, die bereits offen über dem langen grauen Balken mit 7 Löchern aus dem EdCreate-Kit schwebt, der das gefährliche Material darstellt.
- Weitere Informationen zu diesem Programmierprojekt finden Sie im EdCreate-Lehrhandbuch unter <https://meetiedison.com/content/EdCreate/EdCreate-teachers-guide.pdf>.
- Es gibt keine "richtige" Antwort, und die Lösungen werden je nach den von Schülern geschaffenen Einrichtungen variieren. Eine Beispiellösung sieht jedoch wie folgt aus: <https://www.edscratchapp.com?share=rDGRoBba>

Aktivität U4-2.5f: Brieftauben

Umfang der Tätigkeit	Projekt
Empfehlung	Empfohlen als Höhepunkt der Aktivitäten nach Abschluss der Aktivitäten von Einheit 4 "Let's Explore"
Benötigte Materialien	Basissatz, Arbeitsblatt U4-2.5f, Zubehör zur Erstellung des Situationsaufbaus

Überblick

Diese recht unstrukturierte Herausforderung verlangt von den Schülern sowohl die physische Einrichtung als auch die Programmierlösung zu schaffen, um ihre Roboter dazu zu bringen, sich "wie Brieftauben zu verhalten" und ihren Weg "nach Hause" zu finden. Die Anweisungen auf dem Arbeitsblatt sind absichtlich etwas vage. Es gibt nur wenige festgelegte Einschränkungen, und es wird nur ein minimales Maß an Anleitung gegeben. Diese Herausforderung soll den Schülern lediglich einen Ausgangspunkt bieten, von dem aus sie mit ihrer Kreativität, ihrem Einfallsreichtum und ihren rechnerischen Denkfähigkeiten erkunden können.

Tipps und Tricks

- Diese Aktivität funktioniert am besten in Gruppen.
- Dies ist KEINE "ausfallsichere" Aktivität. Die physischen Beschränkungen des Roboters, der Entwurf der Einrichtung, die Präzision der Programmierlösungen und andere damit zusammenhängende Faktoren bedeuten, dass selbst die beste Lösung möglicherweise nicht "perfekt" funktioniert. Ermutigen Sie die Schülerinnen und Schüler, sich mit diesen Einschränkungen auseinanderzusetzen und die beste Lösung auszuloten, die sie finden können. Dies kann als Gelegenheit genutzt werden, um zu diskutieren, wie in der realen Welt Situationen häufig "chaotisch" sind, mit Faktoren, die Kompromisse erfordern. Bei Anwendungen in der realen Welt müssen wir oft „so gut wie möglich“ Lösungen entwickeln, die vielleicht nie zu 100% perfekt sind.
- Programmierlösungen hängen stark von dem von den Schülern erstellten physischen Setup-Design ab. Wenn die Schüler erklären, wie ihre Programmierlösungen und physischen Entwürfe zusammenwirken, ist das eine großartige Übung im Denken durch reale physische Computer.

Einheit 5: Vielseitige Variablen

Tauchen Sie in die wichtigste Funktion von Variablen, Daten und Ausdrücken ein und wenden Sie dabei Gelerntes aus früheren Einheiten an. Schüler runden ihre Erforschung von EdScratch in dieser Einheit ab. Frühere Konzepte werden überdacht und erweitert, um die zusätzliche Flexibilität zu nutzen, die durch die Einbeziehung von Variablen und Operatoren zur Verwaltung von Daten innerhalb ihrer Programme geboten wird.

Lernziele

Schüler werden:

- in die Informatik-Grundlagen von Variablen und Daten eingeführt werden
- untersuchen, wie Daten und Variablen mit Edison verwendet werden können, indem neue Blöcke in den Kategorien "Daten" und "Operatoren" in EdScratch verwendet werden
- Edisons Sensoren und Eingabefähigkeiten mit Hilfe von Variablen und Operatoren weiter erforschen, um das Verhalten in Arbeitsprogrammen zu verfeinern und zu modifizieren
- Anwendung von Informatik-Grundlagen wie Tracing und Debugging, um Projekte unter Verwendung von Abtastung und Variablen abzuarbeiten

Kernkompetenzen: Variablen und Speichern von Werten, Daten, Operatoren und Berechnungen in Programmen, Rückverfolgung (Tracing).

Lektionen und Aktivitäten in dieser Einheit

Diese Einheit umfasst eine Lektion mit insgesamt vier Basisaktivitäten und acht Erweiterungsaktivitäten.

Lektion 1: Mathematik und Daten in EdScratch

- [U5-1.1 Wir erforschen Ausdrücke](#)
- [U5-1.2 Wir erforschen Edisons Lichtsensoren](#)
 - o [U5-1.2a Edison, die Motte](#)
 - o [U5-1.2b Herausforderung angenommen: Edison, die Kakerlake](#)
- [U5-1.3 Variablen untersuchen](#)
 - o [U5-1.3a Herausforderung angenommen: Spiralförmige Spinnenfalle](#)
 - o [U5-1.3b Ein zufälliges Quadrat fahren](#)
- [U5-1.4 Die Verwendung von Variablen mit Sensordaten untersuchen](#)

- o [U5-1.4a Herausforderung angenommen: Edison, der Sprinter](#)
- o [U5-1.4b Edison - gesteuerte Fahnenmaschine](#)
- o [U5-1.4c Hey Edison, wo soll ich hin?](#)
- o [U5-1.4d Der Edison-Chor](#)

Lektion 1: Mathe und Daten in EdScratch

Diese Lektion, die die letzten unerforschten Abschnitte der EdScratch-Sprache einführt, konzentriert sich auf Daten. Während blockbasierte Programmiersprachen den Benutzern oft die Notwendigkeit nehmen, sich in die Details der Daten in der Programmierung zu vertiefen, ist die Entwicklung eines grundlegenden Arbeitsverständnisses dieser grundlegenden Informatikkomponente ein entscheidender Teil der Entwicklung der digitalen Kompetenz von Schülern und bei der Vorbereitung auf textbasierte Programmiersprachen.

Variablen, die durch die spezielle Kategorie "Daten" in EdScratch erstellt und verwendet werden können, werden in dieser Lektion vorgestellt und dann untersucht. Schüler lernen, wie Variablen zum Speichern von Werten verwendet werden. Schüler experimentieren mit der Verwendung und Manipulation von Daten in funktionierenden EdScratch-Programmen, einschließlich der Verwendung von Daten, die mit Edisons eingebauten Sensoren gesammelt wurden. Das Konzept der Verwendung von Mathematik in der Programmierung, einschließlich der wichtigen Praxis der Verwendung von Ausdrücken, wird zusammen mit den verwandten Blöcken aus der Kategorie "Operatoren" in EdScratch eingeführt. Auch die Praxis der Codeverfolgung wird erforscht und geübt.

Diese Lektion umfasst insgesamt vier Basisaktivitäten und acht Erweiterungsaktivitäten:

- [U5-1.1 Wir erforschen Ausdrücke](#)
- [U5-1.2 Wir erforschen Edisons Lichtsensoren](#)
 - o [U5-1.2a Ändern Sie es nach oben: Edison, die Motte](#)
 - o [U5-1.2b Herausforderung angenommen: Edison, die Kakerlake](#)
- [U5-1.3 Lassen Sie uns Variablen untersuchen](#)
 - o [U5-1.3a Herausforderung angenommen: Spiralförmige Spinnenfalle](#)
 - o [U5-1.3b Ändern Sie es nach oben: Ein zufälliges Quadrat fahren](#)
- [U5-1.4 Lassen Sie uns die Verwendung von Variablen mit Sensordaten untersuchen](#)
 - o [U5-1.4a Herausforderung angenommen: Edison, der Sprinter](#)
 - o [U5-1.4b Ändern Sie es nach oben: Edison - gesteuerte Fahnenmaschine](#)
 - o [U5-1.4c Ändern Sie es nach oben: Hey Edison, wo soll ich hin?](#)
 - o [U5-1.4d Wechseln Sie ihn hoch: Der Edison-Chor](#)

Eine besondere Anmerkung zu dieser Lektion

Das Hauptaugenmerk dieser Lektion liegt auf Variablen und Daten, zwei der grundlegend wichtigsten Teile von Allzweck-Programmiersprachen. Es ist nicht ungewöhnlich, dass Schüler (und Dozenten!) das Gefühl haben, dass es zwischen früheren Einheiten und dem Inhalt dieser Lektion eine gewisse Distanz gibt, besonders wenn Sie zum ersten Mal mit Variablen arbeiten. Das Erlernen von Variablen und Mathematik in der Programmierung ist ein wichtiger Schritt in einer sinnvollen Informatikausbildung. Genau wie bei allem, was brandneu ist, kann es einige Zeit dauern, bis man sich mit Mathematik und Variablen in Programmen vertraut gemacht hat. Ermutigen Sie die Schüler dazu, beim Durcharbeiten der Aktivitäten Geduld mit sich selbst zu haben. Sie werden bald sehen, welches kreative Potenzial diese neuen Fähigkeiten im Programmieren freisetzen!

Aktivität U5-1.1 Erforschen wir Ausdrücke

Umfang der Tätigkeit	Niedrig
Empfehlung	
Benötigte Materialien	<ul style="list-style-type: none"> - Arbeitsblatt U5-1.1 - Fakultativ: Programmiergeräte zum Betrachten von 'Operatoren' in EdScratch

Überblick

Diese Offline-Aktivität führt in das Konzept von Ausdrücken in der Codierung ein und demonstriert, wie Berechnungen innerhalb von Ausdrücken verwendet werden können. Die Tatsache, dass alle Computersprachen Zahlen und Mathematik verwenden, wird ebenfalls diskutiert. Durch die Entwicklung eines Verständnisses dafür, wie Ausdrücke und Mathematik offline funktionieren, werden die Schülerinnen und Schüler in die Lage versetzt, sich mit komplexeren rechnergestützten Denkweisen auseinanderzusetzen, so dass sie diese Elemente mit Zuversicht in EdScratch-Programmen in späteren Aktivitäten verwenden können.

Viele Menschen haben die Befürchtung, dass die Computerprogrammierung ein hohes Maß an komplizierter Mathematik erfordert. Für viele Programmiersprachen, einschließlich EdScratch, ist dies einfach nicht wahr. Die Verwendung von Ausdrücken ist jedoch erforderlich, um einige Dinge in EdScratch zu tun und einige von Edisons Sensoren auf sinnvolle Weise zu verwenden. Gleichermaßen ermöglicht es die Fähigkeit, grundlegende Mathematik, wie z.B. Arithmetik, innerhalb von EdScratch-Programmen zu verwenden, interessantere Dinge mit den Robotern zu tun und Edisons Sensoren und Daten auf kreativere und aufregendere Weise zu nutzen.

Auch die Idee der Nachverfolgung wird in dieser Aktivität eingeführt. Die formale Praxis der Ablaufverfolgung besteht darin, Zeile für Zeile durch ein Programm zu gehen und wichtige Werte aufzuzeichnen. Code-Ablaufverfolgung wird oft durchgeführt, um das Debuggen von Code zu erleichtern, kann aber auch nützlich sein, wenn Sie einfach nur verstehen müssen, was in einem Programm geschieht oder was passieren wird, wenn eine Eingabe einen anderen Wert hat. Während die Notwendigkeit, Werte in EdScratch aufzuzeichnen, begrenzt ist, macht diese Aktivität die Schüler mit dem Aufzeichnen von Berechnungen vertraut und ermöglicht es ihnen die Gewohnheit zu entwickeln, in zukünftigen Aktivitäten Programme zu verfolgen.

Tipps und Tricks

- Binär wird in den EdScratch-Lektionen nicht unterrichtet. In dieser Aktivität wird die Tatsache erwähnt, dass Sie „ohne eine Computersprache jeden einzelnen Befehl mit nichts als 1er und 0er schreiben müssten“, was ein Verweis auf die Tatsache ist, dass alle Computer mit Binärzeichen arbeiten. Wenn Sie binär unterrichten, sollten Sie in Erwägung ziehen, eine separate Lektion über Binärprogramme in diese Aktivität einzubinden.
- Wenn es um Ausdrücke und jegliche Mathematik in EdScratch-Programmen geht, ist es wichtig zu beachten, dass Edison nur mit ganzen Zahlen umgehen kann. Gleichermaßen kann Edison nur in "echter" Mathematik arbeiten. Wenn Schüler Berechnungen eingeben, die z.B. durch Null dividieren, können sie Fehler erhalten, oder der Roboter kann sich auf unerwartete Weise verhalten.
- Auch wenn es im Arbeitsblatt nicht erforderlich ist, können Sie die Schülerinnen und Schüler die Kategorie "Operatoren" in EdScratch betrachten lassen, um die sechs Blöcke zu finden, die in Ausdrücken verwendet werden, und damit beginnen, zusammensetzen, wie die Blöcke in EdScratch-Programmen verwendet werden können.

Lösungsschlüssel

Frage	typ	Answer	Hinweis
1	EA	<i>Bedeutung: Ist $2 * 2$ dasselbe wie 4? Ergibt: Wahr</i>	
2	EA	<i>Bedeutung: Ist 2 größer oder gleich 4? Ergibt: Falsch</i>	
3	EA	<i>Bedeutung: Ist $2 + 2$ nicht gleich 4? Ergibt: Falsch</i>	
4	EA	<i>Bedeutung: Ist $2 - 1$ weniger als $4 - 1$? Ergibt: Falsch</i>	

Aktivität U5-1.2 Erforschen wir Edisons Lichtsensoren

Umfang der Tätigkeit	Groß
Empfehlung	- Beenden Sie die Aktivität U4-2.2 vor dieser Aktivität - Beenden Sie die Aktivität U5-1.1 vor dieser Aktivität
Benötigte Materialien	Basissatz, Arbeitsblatt U5-1.2

Überblick

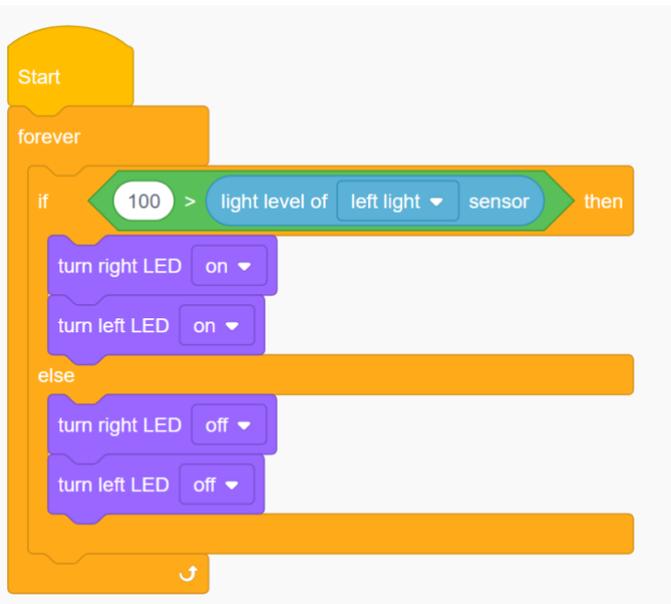
Diese Aktivität stellt Edisons Sensoren für sichtbares Licht vor, erklärt, wie die Sensoren funktionieren und erforscht dann die Verwendung der Sensoren mit Ausdrücken, um zwei verschiedene EdScratch-Programme zu erstellen. Um Edisons Sensoren für sichtbares Licht, die sichtbares Licht erkennen und das Ergebnis als Lichtmesswert an Edison zurückmelden, effektiv nutzen zu können, müssen Schüler die Konzepte von Ereignissen und Bedingungen verstehen und in der Lage sein, Ausdrücke in EdScratch-Programmen zu verwenden.

Mehr als bei jedem anderen Sensor veranschaulicht die Verwendung der Lichtsensoren die Idee, dass Sensoren Daten erzeugen und dass es diese Daten sind, die vom Roboter verwendet werden. Die Verwendung der Sensoren für sichtbares Licht in Programmen hilft Schülern zu erkennen, dass es "hinter den Kulissen" blockbasierter Programmiersprachen Informationsschichten gibt. Die Entwicklung eines Verständnisses für diese Tatsache hilft den Schülern, sich auf die Arbeit mit kreativeren und komplexeren Programmen in EdScratch vorzubereiten und den Grundstein für den Übergang zu textbasierten Sprachen wie EdPy zu legen.

Tipps und Tricks

- Wenn es um Ausdrücke und jegliche Mathematik in EdScratch-Programmen geht, ist es wichtig zu beachten, dass Edison nur mit ganzen Zahlen umgehen kann. Ebenso kann Edison nur in "echter" Mathematik arbeiten. Wenn Schüler also Berechnungen durchführen, die z.B. durch Null dividieren, können sie Fehler erhalten, oder der Roboter kann sich auf unerwartete Weise verhalten.
- Wenn ein Operator-Eingabeparameter in einem EdScratch-Programm leer gelassen wird, wird er vom Roboter als '0' (Null) gelesen.
 - Der Operator kann in diesem Fall NaN im Eingabeparameterraum anzeigen, was für 'keine Zahl' steht.
- Edison wird das gesamte sichtbare Umgebungslicht erkennen und ist nicht in der Lage, die Lichtquelle zu bestimmen. Berücksichtigen Sie, wie hell oder dunkel der Raum ist, wenn Sie Programme mit Edisons Lichtsensoren testen.
- Ermutigen Sie die Schüler, verschiedene Werte in den von ihnen verwendeten Ausdrücken mit Lichtsensorwerten zu testen, um herauszufinden, was mit ihrem Roboter in der Umgebung, in der sie ihr Programm ausführen, am besten funktioniert.

Lösungsschlüssel

Frage	Typ	Antwort/Beispielantwort	Hinweis
1	RC	 <pre> Start forever if (100 > light level of left light sensor) then turn right LED on turn left LED on else turn right LED off turn left LED off </pre>	

Aktivität U5-1.2a : Die Motte Edison

Umfang der Tätigkeit	Mittel
Empfehlung	Beenden Sie die Aktivität U5-1.2 vor dieser Aktivität
Benötigte Materialien	Basissatz, Arbeitsblatt U5-1.2a, Taschenlampen

Überblick

Schüler erforschen, wie sie Edisons Lichtsensoren zusammen verwenden können, um ein Programm für Edison zu erstellen, mit dem der Roboter phototropes Verhalten nachahmt. Die Tatsache, dass die Lichtsensoren Lichtmesswerte als numerische Werte speichern, so dass diese Werte mit Hilfe von Operatoren in einem Ausdruck miteinander verglichen werden können, wird in dieser Aktivität demonstriert.

Tipps und Tricks

- Sie benötigen eine Taschenlampe und eine flache Oberfläche, die von anderen hellen Lichtquellen, wie Sonnenlicht oder Leuchtstoffröhren, entfernt ist.
- Sobald Edison die helle Lichtquelle "sieht", wird der Roboter auf sie zufahren. Indem Sie die Taschenlampe bewegen, können Sie steuern, wohin Edison fährt.
- Wenn die Umgebung heller als die Lichtquelle ist, kann Edison diese nicht erkennen und reagiert nicht wie erwartet.
- Das "Motten"-Programm in dieser Aktivität sollte auf die gleiche Weise funktionieren wie das Programm, das durch das Strichcode-Programm „follow a torch“ aktiviert wird. Sie können sich dafür entscheiden, dass Schüler diesen Strichcode erneut besuchen, um zu sehen, wie ihr Programm im Vergleich zum Strichcode-Programm funktioniert.
- Eine Programmierlösung ist unter <https://www.edscratchapp.com?share=80vEnJYd> zu finden.

Aktivität U5-1.2b : Edison, die Kakerlake

Umfang der Tätigkeit	Mittel
Empfehlung	- Beenden Sie die Aktivität U5-1.2 vor dieser Aktivität - Diese Aktivität baut auf Konzepten auf, die in U5-1.2a näher erläutert werden. Ziehen Sie in Betracht, dies als eine Erweiterung dieser Aktivität zu verwenden.
Benötigte Materialien	Basissatz, Arbeitsblatt U5-1.2b, Taschenlampen

Überblick

Schüler erforschen, wie sie Edisons Lichtsensoren zusammen verwenden können, um ein Programm für Edison zu erstellen, mit dem der Roboter negatives phototropes Verhalten nachahmt. Die Tatsache, dass die Lichtsensoren Lichtmesswerte als numerische Werte speichern, so dass diese Werte mit Hilfe von Operatoren in einem Ausdruck miteinander verglichen werden können, wird in dieser Aktivität demonstriert.

Tipps und Tricks

- Sie benötigen eine Taschenlampe oder andere Lichtquelle und eine ebene Oberfläche, die von anderen hellen Lichtquellen, wie Sonnenlicht oder Leuchtstoffröhren, entfernt ist.
- Sobald Edison die helle Lichtquelle "sieht", sollte der Roboter in die entgegengesetzte Richtung fahren. Indem Sie die Taschenlampe bewegen, können Sie steuern, wohin Edison fährt.
- Wenn die Umgebung heller als die Lichtquelle ist, kann Edison diese nicht erkennen und reagiert nicht wie erwartet.
- Das "Kakerlaken"-Programm ist effektiv dasselbe wie das "Motten"-Programm aus Aktivität U5-1.2b, aber die Logik, wie der Roboter auf Licht reagieren soll, ist umgekehrt.

Lösungsschlüssel

Frage	Typ	Antwort/Beispielantwort	Hinweis
1	SE	<i>Für immer wenn rechter Lichtsensor > linker Lichtsensor nach links gehen sonst nach rechts gehen</i>	

Aktivität U5-1.3 Wir untersuchen die Variablen

Umfang der Tätigkeit	Groß
Empfehlung	
Benötigte Materialien	Basissatz, Arbeitsblatt U5-1.3

Überblick

Das Konzept der Variablen, eine der grundlegenden Komponenten des Codes, wird in dieser Aktivität eingeführt. Das Arbeitsblatt erklärt, wie Variablen funktionieren und lässt die Schüler dann ein Programm anhand einer Variable verfolgen, um den Wert dieser Variable an verschiedenen Stellen im Programm zu verstehen. Wenn die Schüler verstehen, wie Variablen funktionieren und wie sich der Wert einer Variable

ändern kann, können sie ihre eigenen Programme unter Verwendung von Variablen erstellen.

Auch wenn EdScratch es Schülern ermöglicht, viele Programme ohne die Verwendung von Variablen zu schreiben, kann der volle Umfang der Sprache nur durch die Verwendung von Variablen erschlossen werden. Erstellt und verwaltet unter Verwendung der Kategorie 'Daten' in EdScratch, funktionieren Variablen anders als andere Blöcke in der Sprache. Wenn Sie sich die Zeit nehmen, diese etwas erklärungsintensive Aktivität durchzuarbeiten, wird es den Schülern helfen, die Erstellung und Verwendung von Variablen in sinnvollen Programmen zu beherrschen.

Tipps und Tricks

- Nur Variablennamen mit zulässigen Zeichen können zusammengestellt und an Edison geschickt werden. Wenn Schüler Fehlermeldungen erhalten, wenn sie versuchen, ein Programm mit Variablen herunterzuladen, überprüfen Sie den Namen der Variablen.
- Für eine Auffrischung der Verwendung von Edisons 'set motors' Blöcken, siehe Aktivität U2-2.5 „Wir untersuchen Edisons Motoren“.
- Binär wird in diesen Lektionen nicht gelehrt, und daher sind die "Bitverschiebungs"-Blöcke nicht in den Arbeitsblättern der Schüler enthalten. Wenn Sie Binär unterrichten, könnten Sie die "Bitverschiebungs"-Blöcke in einer Erweiterungslektion über Variablen und Binär verwenden.
- Wenn es um Ausdrücke und jegliche Mathematik in EdScratch-Programmen geht, ist es wichtig zu beachten, dass Edison nur mit ganzen Zahlen umgehen kann. Ebenso kann Edison nur in "echter" Mathematik arbeiten. Wenn Schüler also Berechnungen durchführen, die z.B. durch Null dividieren, können sie Fehler erhalten, oder der Roboter kann sich auf unerwartete Weise verhalten.
- Wenn ein Operator-Eingabeparameter in einem EdScratch-Programm leer gelassen wird, wird er vom Roboter als '0' (Null) gelesen.
 - o Der Operator kann in diesem Fall NaN anzeigen, was für 'not a number' steht.
- Wenn Schüler eine Variable oder einen Operatorblock innerhalb eines „Warte“- (wait) Blocks verwenden möchten, müssen sie den "Warte () Millisekunden“- (wait milliseconds) Block verwenden. (Der andere Warte-Block kann keine blasenförmigen Blöcke als Eingabeparameter verwenden). Der Grund dafür, dass dieser Block in Millisekunden statt in Sekunden angegeben wird, hat mit einer Grenze zu tun, wie Edison die Mathematik berechnen kann. Im Wesentlichen weiß Edison nicht, was Dezimalstellen

sind. Daher werden 0,3 Sekunden, wenn sie in Edison gespeichert werden, gerundet und zu 0 Sekunden. Das macht jede Art von Mathematik äußerst ungenau. Wenn wir Millisekunden verwenden, können wir 0,3 Sekunden als 300 Millisekunden darstellen, was eine Zahl ist, die Edison versteht. Auf diese Weise können Berechnungen ohne grobe Rundungsfehler durchgeführt werden.

- o Die anderen Eingaben, die in EdScratch in Sekunden erfolgen, können Dezimalstellen in den Eingabeparametern akzeptieren. Der Compiler, der den EdScratch-Code konvertiert, bevor er an Edison gesendet wird, konvertiert alle Zeiteingaben in diesen Blöcken in Millisekunden, bevor die Werte an Edison gesendet werden.
- Wenn Ihre Schülerinnen und Schüler bereits grundlegende Algebrakenntnisse haben, kann die Wertetabelle in diesem Arbeitsblatt ausgefüllt werden, indem sie sich die drei Spalten als n , $n \cdot 200$ und $n+1$ vorstellen.
- Aufgrund geringfügiger mechanischer Unterschiede bei den Motoren und Encodern in den verschiedenen Edison-Robotern müssen die Schülerinnen und Schüler möglicherweise die Eingabeparameter des Antriebsblocks, die sich auf die Drehung in diesem Programm beziehen, an ihre Roboter anpassen. Um die Genauigkeit zu verbessern, können die Schüler auch entweder einen "Stoppmotoren"-Block oder einen "Warte"-Block mit einem Sortier-Eingabewert (z.B. .2 Sekunden) zwischen den Antriebsbefehlen hinzufügen (auch am unteren Ende des Codes innerhalb der Schleife).
- Wenn Schüler Schwierigkeiten haben, das Muster zu sehen, das Edison bei der Ausführung dieses Programms erzeugt, versuchen Sie, einen Stift an Edison anzubringen, so dass der Roboter die Form "zeichnet", während er sich bewegt. Um das Anbringen eines Stiftes schnell und einfach zu machen, sollten Sie die Herstellung eines Stifthalters mit 3D-Druck in Erwägung ziehen, wie z.B. denjenigen, der unter <https://www.thingiverse.com/thing:2949946> erhältlich ist.

Lösungsschlüssel

Frage	typ	Antwort/Beispielantwort	Hinweis																								
1	EA	<p>In Schleife # Anfangswert von DriveLength Wait-Block-Eingangswert (in Millisekunden) Neuer Wert von DriveLength</p> <table style="margin-left: auto; margin-right: auto;"> <tr> <td>1</td> <td>1</td> <td>200</td> <td>2</td> </tr> <tr> <td>2</td> <td>2</td> <td>400</td> <td>3</td> </tr> <tr> <td>3</td> <td>3</td> <td>600</td> <td>4</td> </tr> <tr> <td>5</td> <td>5</td> <td>1000</td> <td>6</td> </tr> <tr> <td>7</td> <td>7</td> <td>1400</td> <td>8</td> </tr> <tr> <td>10</td> <td>10</td> <td>2000</td> <td>11</td> </tr> </table>	1	1	200	2	2	2	400	3	3	3	600	4	5	5	1000	6	7	7	1400	8	10	10	2000	11	
1	1	200	2																								
2	2	400	3																								
3	3	600	4																								
5	5	1000	6																								
7	7	1400	8																								
10	10	2000	11																								

2	SE	<i>Dieses Programm lässt Edison in einer rechtwinkligen Spirale fahren, die aus der Mitte herauswächst.</i>	Die Schüler sollten beachten, dass der Roboter in einem nach außen spiralförmigen Muster fährt.
3	SE	<i>Der Code verwendet eine Variable, die sich bei jeder Schleifenwiederholung um 1 erhöht. Da die Variable mit 200 multipliziert wird und dies der Wert des 'Warte'-Blocks wird, wird die 'Warte'-Zeit mit jeder Schleifenwiederholung länger. Das bedeutet, dass der Roboter am Ende in jeder Schleife länger fährt, was dazu führt, dass der Roboter in einer nach außen gerichteten Spirale fährt.</i>	Die Schüler sollten erkennen, dass der steigende Wert der Variablen die Zeit verlängert, die die Motoren "warten", während sie "auf Vorwärtsfahrt" eingestellt sind, was die Distanz jedes gefahrenen Beins erhöht.

Aktivität U5-1.3a: Spiralförmige Spinnenfalle

Umfang der Tätigkeit	Mittel (Haupttätigkeit) Groß (Haupttätigkeit + zusätzliche technische Herausforderung)
Empfehlung	Beenden Sie die Aktivität U5-1.3 vor dieser Aktivität
Benötigte Materialien	- Basissatz, Arbeitsblatt U5-1.3a - Optional: Zubehör für die Herstellung der 'Spinnenseide', Arbeitsblatt U5-1.3 (als Referenz)

Überblick

Aufbauend auf den Programmkonzepten, die in Aktivität U5-1.3 untersucht wurden, fordert diese Erweiterungsaktivität die Schüler auf, sich das "Spiral-Out"-Programm aus Aktivität U5-1.3 stattdessen als ein "Spiral-In"-Programm vorzustellen. Diese Aktivität, die als Sprungbrett für die Schülerinnen und Schüler gedacht ist, um ihnen zu helfen, die Verwendung von Variablen und Berechnungen in selbst erstellten Programmen zu erforschen, fordert die Schülerinnen und Schüler auf, die Logik des "Spiral-Out"-Programms zu durchdenken, um seine Hauptkomponenten zu modifizieren und erneut anzuwenden.

Eine zusätzliche "Mini-Herausforderung", die mit dieser Aktivität verbunden ist, bietet den Schülern die Möglichkeit, Computerprogrammierung und physikalische Technik zu vermischen, um ihre Roboter in Fallen legende, spiralförmige Spinnen zu verwandeln.

Tipps und Tricks

- Vielleicht möchten Sie das "Spiral-Out"-Programm aus Aktivität U5-1.3 noch einmal durchgehen, um Schülern zu helfen, ihre "Spiral-In"-Programme auszuarbeiten.
- Erinnern Sie die Schüler daran, gute Variablennamen zu verwenden und illegale Zeichen im Namen ihrer Variablen zu vermeiden.
- Einen Weg zu finden, die "Spinnenseide" zu verlassen UND dafür zu sorgen, dass diese "Seide" an Ort und Stelle bleibt (anstatt vom Roboter bei seiner Bewegung mitgerissen und aus der Form gebracht zu werden), ist keine ausfallsichere Aktivität. Viele Faktoren, darunter die Art der Materialien, die die Schülerinnen und Schüler verwenden, die Art und Weise, wie sie die Schnur (oder eine andere Substanz) an dem Roboter und der Fahrfläche befestigen (oder nicht befestigen), die Geschwindigkeit, mit der der Roboter fährt, usw., beeinflussen, wie erfolgreich sie bei dieser Bonusaktivität sind. Es kann sein, dass Schülerinnen und Schüler nicht in der Lage sind, eine Möglichkeit zu schaffen, diese zusätzliche Herausforderung "perfekt" zu bewältigen. Das ist in Ordnung! Die Schlüsselerfahrungen werden sich aus dem Prozess des Experimentierens und der Problemlösung ergeben, nicht aus dem Endergebnis.
 - Wenn die Schülerinnen und Schüler die Form ihres Roboters sehen wollen, aber die Schnur überspringen möchten, sollten sie in Erwägung ziehen, stattdessen einen Stift am Roboter anzubringen.
- Ein Beispiel für eine Programmierlösung kann unter <https://www.edscratchapp.com?share=rDBEwzbR> eingesehen werden.

Aktivität U5-1.3b : Eine zufällige Quadratform fahren

Umfang der Tätigkeit	Klein
Empfehlung	Beenden Sie die Aktivität U5-1.3 vor dieser Aktivität
Benötigte Materialien	Basissatz, Arbeitsblatt U5-1.3b

Überblick

Bei dieser Aktivität wird eine Variable mit dem "zufälligen" Zahlenblock verwendet, so dass der Wert der Variable bei jeder Programmausführung zufällig gesetzt wird. Diese Aktivität demonstriert eine andere Art und Weise der Verwendung von Variablen in Programmen in EdScratch, die nur eingestellte Werte verwenden (im Gegensatz zu Sensordaten) und bietet Schülern eine weitere Möglichkeit, das Rechnen innerhalb eines Programms zu üben.

Die zusätzliche "Mini-Herausforderung" bringt die Schüler dazu, darüber nachzudenken, wie Daten im Programm gespeichert werden, und fordert sie dazu heraus, eine zweite Möglichkeit zur Verwendung dieser Daten im Programm zu schaffen, indem sie Edison dazu bringen, den Wert zu "signalisieren".

Tipps und Tricks

- Wenn Schüler die Mini-Herausforderung abschließen, sollten Sie sie vielleicht daran erinnern, dass eine Variable in einem Programm mehrfach verwendet werden kann.
- Eine Beispielprogrammierlösung für die Mini-Challenge ist unter <https://www.edscratchapp.com?share=7bxNWWYZ> zu sehen.

Lösungsschlüssel

Frage	typ	Answer	Hinweis
1	EA	<p>Wenn Zufallszahl: Seitenlänge des Quadrats:</p> <p>1 500</p> <p>2 1000</p> <p>3 1500</p> <p>4 2000</p> <p>5 2500</p> <p>6 3000</p>	

Aktivität U5-1.4 Die Verwendung von Variablen mit Sensordaten

Umfang der Tätigkeit	Mittel
Empfehlung	
Benötigte Materialien	- Basissatz, Arbeitsblatt U5-1.4, Tätigkeitsblatt U5-1 - Fakultativ: Zubehör zur Markierung eines Testbereichs

Überblick

Diese Aktivität stellt vor, wie Variablen und Sensoren in Kombination verwendet werden können. Es gibt zwei Hauptmöglichkeiten: Die Daten von Sensoren können in einer Variablen gespeichert werden, und Sensorereignisse können dazu verwendet werden, Variablen zu beeinflussen. Das Programmierbeispiel in dieser Aktivität konzentriert sich auf die zweite Art. Das Programm inkrementiert eine Variable jedes Mal, wenn ein Ereignis erkannt wird. Schüler sehen diese Fähigkeit in Aktion, indem sie lesen und dann ein Programm ausführen, das Edison zum Fahren bringt und die schwarzen Linien zählt.

Obwohl in dieser Aktivität kein explizites Beispiel gezeigt wird, ist der Gedanke wichtig, dass der Wert eines Sensors in einer Variablen gespeichert werden kann. Variablen können vom Programmierer auf bestimmte Werte eingestellt werden, können aber auch durch andere Eingaben, wie z.B. einen Sensormesswert, gesetzt werden.

Tipps und Tricks

- Die Aktivität U5-1.4a ist eine Erweiterung des Beispielprogramms in dieser Aktivität. Sie können sich dafür entscheiden, dass Schüler diese zusätzliche Programmieraufgabe ausprobieren, während sie ihren Testbereich für diese Aktivität eingerichtet haben.
- Die Fragen 3 und 4 auf dem Arbeitsblatt fordern die Schüler auf, ihre eigene Idee für die Verwendung von Sensordaten in einer Variablen in einem funktionierenden EdScratch-Programm zu entwickeln. Wenn Sie diese Aktivität kürzer halten wollen, können Sie diese Fragen als Klassendiskussion verwenden, anstatt die Schüler an ihren eigenen Ideen und dem Programmdesign arbeiten zu lassen.

Frage	Typ	Antwort/Beispielantwort	Hinweis
1	EA	<i>Dieses Programm weist Edison zunächst an, den Line Tracker einzuschalten und die Variable LineCount auf Null zu setzen. Dann weist es den Roboter an, vorwärts zu fahren, bis LineCount gleich vier ist. Es gibt ein Interrupt-"Ereignis" in diesem Programm mit einer Unterroutine, die besagt, dass jedes Mal, wenn Edison eine nicht reflektierende (schwarze) Linie erkennt, der Roboter den Wert von LineCount inkrementieren (um 1 addieren) soll. Sobald dieses Ereignis 4 Mal ausgelöst wurde, wird die "bis"-Bedingung erfüllt und der Roboter hört auf zu fahren.</i>	In den Antworten der Schüler sollten die in der Antwort/Beispielantwort vermerkten Kernkompetenzen angegeben werden.
2	SE	<i>Ich denke, dass die Variable 'LineCount' genannt wird, weil die Information, die diese Variable speichert, sich darauf bezieht, wie viele schwarze Linien der Roboter erkannt oder 'gezählt' hat.</i>	Die Antworten der Schüler sollten erkennen lassen, dass die vom Roboter erkannten Linien "gezählt" werden, indem diese Variable zur Speicherung der laufenden Zählung verwendet wird.
3	SE	<i>Meine Idee ist es, Edisons Hinderniserkennung zu verwenden, um zu verfolgen, wie viele Leute in mein Zimmer kommen, wenn ich nicht zu Hause bin. Ich kann die Anzahl der erkannten Hindernisse (d.h. Personen, die an Edison vorbeigehen) in einer Variablen speichern, wobei ich den Wert für jedes erkannte Hindernis um 1 erhöhe. Dann kann ich das Programm dazu veranlassen, beim Drücken des runden Knopfs eine Anzahl von Piepstönen zu geben, die dieser Variable entspricht und mir sagt, wie viele Personen hereingekommen sind.</i>	

4	SE	<p><i>Ich war in der Lage, mein Programm zu schreiben, aber es brauchte einige Versuche.</i></p> <p>https://www.edscratchapp.com?share=RDaGLE0V <i>Meine erste Version hatte nicht die "forever"-Schleife mit dem Hauptprogramm, so dass es sehr schnell endete und nicht funktionierte. Mir wurde auch klar, dass ich die Variable zurücksetzen musste, nachdem ich den Detektor überprüft hatte, damit er nicht einfach immer wieder alle neuen Hindernisse zu den alten hinzufügte. Beim Testen ist mir aufgefallen, dass manchmal nur eine Person, die vor Edison läuft, als mehrere Hindernisse erkannt wurde, also ist dieses System nicht perfekt, aber es funktioniert!</i></p>	<p>Es ist möglich, dass Schüler mit einer Idee gekommen sind, die nicht verwirklicht werden kann.</p>
---	----	---	---

□

Lösungsschlüssel

Aktivität U5-1.4a : Edison, der Sprinter

Umfang der Tätigkeit	Mittel
Empfehlung	Beenden Sie die Aktivität U5-1.4 vor dieser Aktivität
Benötigte Materialien	<ul style="list-style-type: none"> - Basissatz, Arbeitsblatt U5-1.4a, Tätigkeitsblatt U5-1 - Fakultativ: Zubehör zur Markierung eines Testbereichs

Überblick

Diese Programmieraufgabe, die als Erweiterungsaktivität für das Programmbeispiel in Aktivität U5-1.4 konzipiert wurde, veranlasst die Schüler, ihr Verständnis von Sensorereignissen und -variablen anzuwenden, um die Roboter dazu zu bringen, Ereignisse zu verfolgen und die Ausgaben entsprechend zu modifizieren. Indem sie die Ausgabe des Roboters jedes Mal ändern, wenn sich der Wert einer Variablen ändert, können die Schüler sehen, dass sich der Wert der Variablen tatsächlich ändert, was dazu beiträgt, die Idee von Daten in Variablen greifbarer zu machen.

Tipps und Tricks

- Um die Geschwindigkeitsänderung deutlich zu sehen, ist es am besten, einen größeren Testbereich zu verwenden. Vielleicht möchten Sie den Schülern auch raten, die Geschwindigkeit jedes Mal um mehr als eine Stufe zu ändern. Beginnen Sie z.B. mit Geschwindigkeit 1 und erhöhen Sie dann von der ersten Linie aus auf Geschwindigkeit 3.
- Diese Aktivität ist eine Erweiterung des Beispielprogramms in Aktivität U5-1.4. Sie können sich dafür entscheiden, dass Schüler beide Programmierherausforderungen ausprobieren, während sie ihren Testbereich einrichten lassen.
- Ein Beispiel für eine Programmierlösung kann unter <https://www.edscratchapp.com?share=Eb1j5aDm> eingesehen werden.

Tätigkeit U5-1.4b : Edison-gesteuerte Flaggenmaschine

Umfang der Tätigkeit	Projekt
Empfehlung	- Beenden Sie die Aktivität U5-1.4 vor dieser Aktivität - Die Aktivitäten U5-1.4b, U5-1.4c und U5-1.4d verwenden alle Variablen zum Speichern und Verwenden von IR-Nachrichtenwerten. Erwägen Sie die Verwendung mindestens einer dieser Variablen.
Benötigte Materialien	Basissatz, Arbeitsblatt U5-1.4b, Hersteller-Platzvorräte zum Erstellen der Flaggen, EdCreate-Sätze und/oder Zubehör zum Anbringen der Flaggen

Überblick

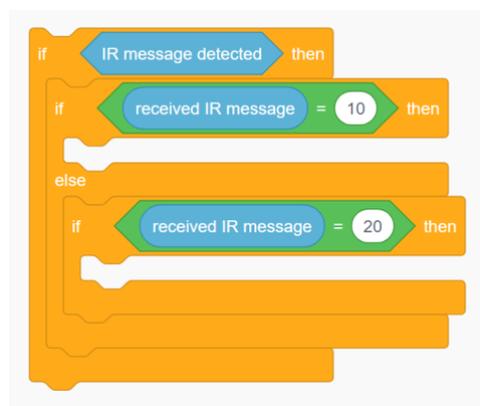
Bei dieser Aktivität geht es darum, Daten von einem der Edison-Sensoren als Wert in einer Variablen zu speichern. Edisons IR-Messaging ermöglicht es mehreren Edisons, Nachrichten zu senden und zu empfangen. Die Roboter können 256 verschiedene Nachrichten senden und erkennen. Diese einzigartigen Nachrichten können verwendet werden, um unterschiedliche Verhaltensweisen im empfangenden Roboter auszulösen, je nachdem, welche Nachricht er erkennt.

Dieses halboffene Projekt fordert die Schülerinnen und Schüler heraus, eine physikalische Computerlösung mit Anwendungen aus der realen Welt zu entwickeln und dabei ihr Verständnis von Codierungskonzepten, einschließlich Bedingungen, Abtastung und Verwendung von Variablen zur Datenspeicherung, anzuwenden. Die

Schüler arbeiten mit zwei Edison-Robotern zusammen, um einen Controller-Bot (Nachrichtensender) und eine Flag-Maschine (Nachrichtempfänger) zu erstellen, die zusammen arbeiten. Der sendende Roboter muss in der Lage sein, eine oder zwei getrennte Nachrichten zu senden, je nachdem, welche Taste gedrückt wird. Der empfangende Roboter muss auf jede dieser Nachrichten unterschiedlich reagieren und je nachdem, welche Nachricht er empfängt, eine unterschiedliche Ausgabe erzeugen.

Ein besonderer Hinweis zu Variablen und IR-Meldungsdaten

Bei der Verwendung von IR-Meldungen in Programmen, die bestimmte Meldungen zum Auslösen verschiedener Ereignisse verwenden, könnten Schüler ihre Programme ohne eine Variable einrichten und nur einen Ausdruck verwenden, um den Wert der IR-Meldung zu überprüfen:



Was das Programmierverfahren betrifft, so ist dies nicht optimal.

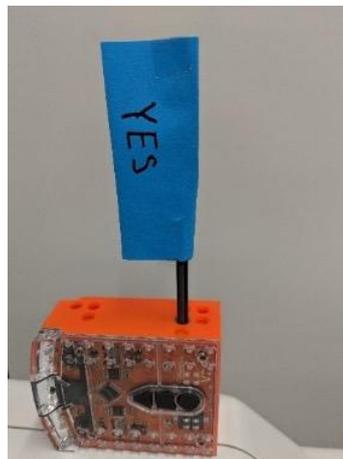
Die Übereinkunft in der Computerprogrammierung besteht darin, beliebige Sensordaten in einer Variablen zu speichern. Wenn Ihr Programm dann mehrere "Wenn"-Prüfungen im Code verwendet, lassen Sie das Programm die Variable mehrmals prüfen, anstatt den Sensor mehrmals zu lesen. Das liegt daran, dass sich der Sensor während der Prüfungen ändern könnte.

Insbesondere bei EdScratch werden Schüler feststellen, dass der 'IR message = 20'-Code im obigen Programm niemals ausgeführt wird. Dies ist darauf zurückzuführen, wie Edison mit den Daten der IR-Nachricht in EdScratch umgeht. Sobald der Roboter die 'if'-Anweisung im 'if-else'-Codeblock verarbeitet, ist der Wert der Nachricht gelesen worden, und der Wert wird aus dem Register gelöscht. Wenn diese Nachricht nicht gleich 10 war, wird der 'if'-Code nicht ausgeführt, aber das IR-Register wird immer noch auf '0' zurückgesetzt. Daher sieht das zweite "wenn" (innerhalb des "dann") immer nur einen Wert von "0" und läuft nie.

Computer, die Sensoren verwenden, löschen das Sensorregister oft auf ähnliche Weise. Dies ist ein hervorragendes Beispiel dafür, warum es viel besser ist, sich daran zu gewöhnen, Variablen zum Speichern von Daten zu verwenden, als sich auf einen direkt gelesenen Sensor zu verlassen.

Tipps und Tricks

- Vielleicht möchten Sie vor dieser Aktivität überprüfen, wie IR-Messaging mit Edison funktioniert, indem Sie die Aktivität U4-2.5 Let's explore messaging with Edison verwenden.
- Schüler steht gerne offen, eine Flag-Maschine auf eine andere Weise als in der Projektbeschreibung in dieser Aktivität zu bauen oder ihre eigene IR-Nachricht zur Erkennung der Erstellung zu entwerfen.
- Eine Variation dieser Aktivität unter Verwendung von TV/DVD-Fernbedienungen ist in Einheit 4, Aktivität U4-2.5a : Ferngesteuerte Flaggenmaschine, enthalten. Sie haben die Wahl, diese beiden Aktivitäten bei der Verwendung einer der beiden zu referenzieren oder sie als zwei Programmierherausforderungen für den Entwurf einer einzigen Konstruktion auszuführen.



- Es gibt keine "richtige" Antwort, aber so kann eine Musterlösung aussehen:
 - Programmgestaltung:
 - <https://www.edscratchapp.com?share=V0eg4KDa>

Lösungsschlüssel

Frage	Typ	Antwort/Beispielantwort	Hinweis
1	SE	<i>Wenn Sie den runden Knopf auf dem Controller-Bot drücken, sendet er die IR-Meldung _10_, und das wird die Flaggenmaschine anweisen, die _'no'_ Seite der Flagge zu zeigen. Wenn Sie den dreieckigen Knopf auf dem Controller-Bot drücken, sendet er die IR-Nachricht _20_, die die Flaggenmaschine anweist, die _'Ja'_ - Seite der Flagge zu zeigen.</i>	Schüler sollten dies nutzen, um sie in ihrem Entwurfsprozess zu unterstützen. Werte und Bewegung sollten sich in ihrem "Flag Machine"-Programm widerspiegeln.

Activity U5-1.4c : Hallo Edison, wo gehe ich hin?

Umfang der Tätigkeit	Projekt
Empfehlung	- Beenden Sie die Aktivität U5-1.4 vor dieser Aktivität - Die Aktivitäten U5-1.4b, U5-1.4c und U5-1.4d verwenden alle Variablen zum Speichern und Verwenden von IR-Nachrichtenwerten. Erwägen Sie die Verwendung mindestens einer dieser Variablen.
Benötigte Materialien	Basissatz, Arbeitsblatt U5-1.4c, Materialien für das Makerspace zur Erstellung des Testraums

Überblick

Bei dieser Aktivität geht es darum, Daten von einem der Edison-Sensoren als Wert in einer Variablen zu speichern. Edisons IR-Messaging ermöglicht es mehreren Edisons, Nachrichten zu senden und zu empfangen. Die Roboter können 256 verschiedene Nachrichten senden und erkennen. Diese einzigartigen Nachrichten können verwendet werden, um unterschiedliche Verhaltensweisen im empfangenden Roboter auszulösen, je nachdem, welche Nachricht er erkennt.

Dieses halboffene Projekt fordert Schülerinnen und Schüler heraus, einen Testraum zu entwerfen und mehrere Programme zu erstellen, die zwei Edison-Roboter dazu bringen, IR-Nachrichten zu verwenden, um ein verzweigtes Labyrinth zu vervollständigen. Die Schülerinnen und Schüler müssen zusammenarbeiten, um ihr Schatzlabyrinth zu entwerfen, ihre beiden Roboter entsprechend ihrem Labyrinth zu programmieren und in der Lage sein, die Sendebefehle zu ändern, um den Roboter an jeden beliebigen Endpunkt des Labyrinths zu bringen. Um ein erfolgreiches Ergebnis in diesem Projekt zu erzielen, müssen die Schülerinnen und Schüler die verschiedenen Konzepte, die sie gelernt haben, einschließlich Sequenz, Bedingungen, Erkennung und Verwendung von Variablen mit Daten kombinieren.

Tipps und Tricks

- Dieses Projekt kann zu zweit durchgeführt werden, ist aber ideal für Gruppen.
- Wenn dies Ihre erste Aktivität ist, bei der Sie Variablen und IR-Meldungsdaten verwenden, lesen Sie bitte den Besonderen Hinweis zu Variablen und IR-Daten im Überblick über die Aktivität U5-1.4b dieses Leitfadens.

- Eine gute Möglichkeit, Schüler nach Abschluss dieses Projekts zu testen, besteht darin, den Schülern den Standort des Schatzes zuzuweisen. Die Schüler sollten dann in der Lage sein, ein erfolgreiches Ergebnis zu demonstrieren, indem sie nur das Programm des Navigationsroboters so ändern, dass er die richtige Abfolge von Nachrichten an den Fahrerroboter sendet. Wenn beide Bots ihre jeweiligen Programme ausführen, sollte der Fahrerroboter den zugewiesenen Zielort erreichen.
- Erinnern Sie die Schüler daran, den empfangenden Roboter seine Programme ausführen zu lassen, bevor der sendende Roboter sein Programm startet.
- Die Programme der Schüler werden stark von ihrem Schatzkartendesign abhängen. Eine Reihe von Beispielprogrammen, die das "Basisdesign" aus dem Arbeitsblatt als Schatzkarte verwenden, sind als Referenz beigefügt:
 - Fahrer-Bot: <https://www.edscratchapp.com?share=RDgmg30B>
 - Navigator-Bot: <https://www.edscratchapp.com?share=RDaGLj0V>
- Wenn in einer Klassenraumumgebung viele Roboter IR-Nachrichten aussenden, haben Sie möglicherweise Probleme mit dem Übersprechen zwischen Gruppen. Eine Möglichkeit, diesem Problem zu begegnen, besteht darin, jeder Gruppe eine begrenzte Anzahl von IR-Codes (fünf sollten je nach Karte des Schülers ausreichen) zuzuweisen, so dass ihr Fahrerroboter nur auf ihren Navigatorroboter hört (und nicht auf andere Roboter). Es folgen Beispielprogramme zur Einrichtung dieser Einrichtung.
 - **Hinweis:** In den folgenden Beispielen ist der erste Schüler die IR-Codes 0-4, der zweite ist 5-9, 10-14, 15-19 und so weiter. Diese Lösung funktioniert für Klassengrößen bis zu 51.
 - Fahrer-Bot: <https://www.edscratchapp.com?share=B0jd35Dg>
 - Navigator-Bot: <https://www.edscratchapp.com?share=wbJv8ebj>
 - **Hinweis:** Die obigen Beispiele beinhalten auch ein 'a Round Button Pressed'-Ereignis, so dass der Fahrer dem Navigator mitteilen kann, wann er startbereit ist.

Lösungsschlüssel

Frage	Typ	Antwort/Beispielantwort	Hinweis															
1	SE	<p><i>Unsere Karte hat drei Wege, die von jeder Kreuzung ausgehen, so dass wir insgesamt 4 IR-Meldungen benötigen:</i></p> <table border="0"> <tr> <td>IR-Nachrichtenwert</td> <td>Edison senden</td> <td>Aktion Edison</td> </tr> <tr> <td>10</td> <td>Navigator</td> <td>Go left</td> </tr> <tr> <td>20</td> <td>Navigator</td> <td>Go straight</td> </tr> <tr> <td>30</td> <td>Navigator</td> <td>Go right</td> </tr> <tr> <td>100</td> <td>Driver</td> <td>Send out next command</td> </tr> </table>	IR-Nachrichtenwert	Edison senden	Aktion Edison	10	Navigator	Go left	20	Navigator	Go straight	30	Navigator	Go right	100	Driver	Send out next command	
IR-Nachrichtenwert	Edison senden	Aktion Edison																
10	Navigator	Go left																
20	Navigator	Go straight																
30	Navigator	Go right																
100	Driver	Send out next command																

Aktivität U5-1.4d: Der Edison-Chor

Umfang der Tätigkeit	Projekt
Empfehlung	<ul style="list-style-type: none"> - Beenden Sie die Aktivität U5-1.4 vor dieser Aktivität - Die Aktivitäten U5-1.4b, U5-1.4c und U5-1.4d verwenden alle Variablen zum Speichern und Verwenden von IR-Nachrichtenwerten. Erwägen Sie die Verwendung mindestens einer dieser Variablen.
Benötigte Materialien	Basissatz, Arbeitsblatt U5-1.4d, Noten oder Zugang für Schüler zum Nachschlagen von Liedern

Überblick

Bei dieser Aktivität geht es darum, Daten von einem der Edison-Sensoren als Wert in einer Variablen zu speichern. Edisons IR-Messaging ermöglicht es mehreren Edisons, Nachrichten zu senden und zu empfangen. Die Roboter können 256 verschiedene Nachrichten senden und erkennen. Diese einzigartigen Nachrichten können verwendet werden, um unterschiedliche Verhaltensweisen in verschiedenen empfangenden Robotern auszulösen.

Dieses halboffene Projekt fordert Schülerinnen und Schüler zur Zusammenarbeit auf, um mehrere Edison-Roboter dazu zu bringen, IR-Nachrichten zu verwenden, um verschiedene Aktionen auszulösen. Dieses Projekt zeigt auf, was bei der Programmierung mit mehreren verschiedenen IR-Nachrichten möglich ist, indem die verschiedenen Nachrichten nicht nur verschiedene Ausgaben auslösen, sondern auch mehrere Roboter Nachrichten prüfen und darauf reagieren lassen. Um ein

erfolgreiches Ergebnis in diesem Projekt zu erzielen, müssen die Schülerinnen und Schüler die verschiedenen Konzepte, die sie gelernt haben, einschließlich Sequenz, Bedingungen, Erkennung und Verwendung von Variablen mit Daten kombinieren.

Tipps und Tricks

- Dieses Projekt benötigt mindestens 3 Edison-Roboter, funktioniert aber hervorragend mit 4+ Robotern, so dass es ideal für Gruppen ist.
- Wenn dies Ihre erste Aktivität ist, bei der Sie Variablen und IR-Meldungsdaten verwenden, lesen Sie bitte den Besonderen Hinweis zu Variablen und IR-Daten im Überblick über die Aktivität U5-1.4b dieses Leitfadens.
- Erwägen Sie, Ihren Schülern einige Liedoptionen oder Notenblätter zur Verwendung in dieser Aktivität zu geben. Gute Optionen für Lieder in einer Runde beinhalten:
 - *Row, Row, Row Your Boat*
 - *Three Blind Mice*
 - *Frère Jacques*
 - *Farmer in the Dell*
- Um das Risiko zu minimieren, dass Roboter IR-Meldungen verpassen, schlagen Sie den Schülern vor, dieses Programm mit den „Chorsänger“-Bots in einem Halbkreis gegenüber dem „Dirigenten“-Bot durchzuführen.
- Wenn mehrere Gruppen Programme nahe beieinander mit denselben Nachrichtenwerten ausführen, kann es zu Übersprechen zwischen den Gruppen kommen.
- Erinnern Sie die Schüler daran, die empfangenden Roboter ihre Programme ausführen zu lassen, bevor der sendende Roboter sein Programm startet.
- Eine ähnliche Programmieraufgabe wie diese Aktivität ist in Einheit 2 in Aktivität U2-2.3a enthalten: Spielen Sie ein Lied in einer Runde. Bei dieser Aktivität verwenden die Schülerinnen und Schüler Warteblocke, um den Start jedes Roboters zu timen. Sie können sich für die Aktivität U2-2.3a entscheiden: Spielen Sie ein Lied in einer Runde und diese Aktivität zusammen, wobei Sie die Vor- und Nachteile jedes Ansatzes hervorheben.
- Das Löschen von IR-Daten mit dem Block "IR-Meldungsdaten löschen", nachdem eine Variable mit dem Wert der IR-Meldung gesetzt wurde, hilft, Probleme bei der Verwendung mehrerer IR-Meldungen in einem Programm zu minimieren, indem alte Daten aus dem Speicher des Roboters gelöscht werden. Ermutigen Sie die Schüler, die IR-Daten als nächste Aktion zu löschen, nachdem der Wert der Variablen mit diesen IR-Daten gesetzt wurde.

- Die Programme der Schüler hängen stark von ihrer Songauswahl ab. Eine Reihe von Beispielprogrammen mit vier Robotern, die "Row, Row, Row Your Boat" spielen, sind als Referenz beigefügt:
 - Dirigenten-Bot: <https://www.edscratchapp.com?share=ZDRedPby>
 - Chorsänger-Bots: <https://www.edscratchapp.com?share=QbOQ3OYk>

Lösungsschlüssel

Frage	Typ	Antwort/Beispielantwort	Hinweis
1	SE	<i>Unsere Chorsänger-Bots spielen 'Row, Row, Row your boat', und jeder neue Bot macht mit, wenn sein Vorgänger die erste Strophe beendet hat.</i>	

2	SE	<p><i>Wir haben drei Sänger-Bots, so dass wir insgesamt 5 IR-Nachrichten benötigen:</i></p> <table border="1"> <thead> <tr> <th>IR-Nachrichtenwert</th> <th>Empfangender Edison</th> <th>sendender Edison</th> <th>Ausführung</th> </tr> </thead> <tbody> <tr> <td>10</td> <td>Leiter</td> <td>Ausführender #1</td> <td>abspielen</td> </tr> <tr> <td>1</td> <td>Ausführender #1</td> <td>Leiter</td> <td>send abspielen</td> </tr> <tr> <td colspan="4">command to Ausführender # 2</td> </tr> <tr> <td>20</td> <td>Leiter</td> <td>Ausführender #2</td> <td>abspielen</td> </tr> <tr> <td>2</td> <td>Ausführender #2</td> <td>Leiter</td> <td>send abspielen</td> </tr> <tr> <td colspan="4">command to Ausführender # 3</td> </tr> <tr> <td>30</td> <td>Leiter</td> <td>Ausführender #3</td> <td>abspielen</td> </tr> </tbody> </table>	IR-Nachrichtenwert	Empfangender Edison	sendender Edison	Ausführung	10	Leiter	Ausführender #1	abspielen	1	Ausführender #1	Leiter	send abspielen	command to Ausführender # 2				20	Leiter	Ausführender #2	abspielen	2	Ausführender #2	Leiter	send abspielen	command to Ausführender # 3				30	Leiter	Ausführender #3	abspielen	
IR-Nachrichtenwert	Empfangender Edison	sendender Edison	Ausführung																																
10	Leiter	Ausführender #1	abspielen																																
1	Ausführender #1	Leiter	send abspielen																																
command to Ausführender # 2																																			
20	Leiter	Ausführender #2	abspielen																																
2	Ausführender #2	Leiter	send abspielen																																
command to Ausführender # 3																																			
30	Leiter	Ausführender #3	abspielen																																

Einheit 6: Erfinder!

Setzen Sie Ihr ganzes Edison- und EdScratch-Wissen in die Tat um! Durch das Entwerfen und Entwickeln eigener Projekte unter Verwendung iterativer Planungs-, Herstellungs- und Testzyklen setzen die Schülerinnen und Schüler die Schlüsselkonzepte des rechnergestützten Denkens, Problemlösens, Programmierens und physikalischen Rechnens um, die sie gelernt haben, um in dieser Kulminationseinheit zu arbeiten.

Lernziele

Schüler werden:

- etwas über den Design-Build-Test-Zyklus und Strategien, wie Zerlegung und iteratives Testen, für die Lösung von Physical-Computing-Problemen lernen
- ihr Verständnis der wichtigsten Prinzipien des rechnergestützten Denkens und der Informatik durch offene Projekte demonstrieren

Kernkompetenzen: der Design-Build-Test-Zyklus, Zerlegung und Problemlösung, iteratives Testen

Lektionen und Aktivitäten in dieser Einheit

Diese Einheit umfasst eine Lektion mit insgesamt zwei Basisaktivitäten und fünf Erweiterungsaktivitäten.

Lektion 1: Entwerfen, Bauen, Testen, Wiederholen

- [U6-1.1 Wir untersuchen den Design-Build-Test-Zyklus](#)
 - o [U6-1.1a Herausforderung angenommen: Erfinde eine imaginäre Kreatur](#)
 - o [U6-1.1b Herausforderung angenommen: Erfinde einen Wattebausch-Werfer](#)
 - o [U6-1.1c Herausforderung angenommen: Einen Einbrecheralarm erfinden](#)
 - o [U6-1.1d Herausforderung angenommen: Eine Mausefalle erfinden](#)
 - o [U6-1.1e Herausforderung angenommen: Einen Kombinationssafe erfinden](#)
- [U6-1.2 Lass uns ein Spukhaus erforschen](#)

Lektion 1: Entwerfen, Bauen, Testen, Wiederholen

Da die gesamte EdScratch-Sprache erforscht wird, konzentriert sich diese Lektion auf die kreative Problemlösung sowohl bei Programmier- als auch bei physikalischen Designherausforderungen. Die Schüler lernen den Design-Build-Test-Zyklus und formale Strategien für die Problemlösung im Physical Computing kennen, einschließlich Zerlegung und iteratives Testen. Diese Fähigkeiten werden dann auf die Probe gestellt, wenn die Schülerinnen und Schüler ihr Wissen über die wichtigsten Prinzipien des rechnergestützten Denkens, Programmierens und der Informatik in offenen Projekten unter Beweis stellen.

Diese Lektion umfasst insgesamt zwei Basisaktivitäten und fünf Erweiterungsaktivitäten:

- [U6-1.1 Wir untersuchen den Design-Build-Test-Zyklus](#)
 - o [U6-1.1a Herausforderung angenommen: Erfinde eine imaginäre Kreatur](#)
 - o [U6-1.1b Herausforderung angenommen: Erfinde einen Wattebausch-Werfer](#)
 - o [U6-1.1c Herausforderung angenommen: Einen Einbrecheralarm erfinden](#)
 - o [U6-1.1d Herausforderung angenommen: Eine Mausefalle erfinden](#)
 - o [U6-1.1e Herausforderung angenommen: Einen Kombinations safe erfinden](#)
- [U6-1.2 Lass uns ein Spukhaus erforschen](#)

Aktivität U6-1.1 Untersuchen wir den Design-Build-Test-Zyklus

Umfang der Tätigkeit	Groß (nur Basisaufgaben) Projekt (mit dem Aufbau)
Empfehlung	
Benötigte Materialien	- Basissatz, Arbeitsblatt U6-1.1, Tätigkeitsblatt U6-1 - Wenn Sie bauen: Herstellzubehör

Überblick

In dieser Lektion wird formell ein strukturierter Prozess zur Erstellung von Roboter-Erfindungen unter Verwendung von Edison-Robotern eingeführt. Während viele der Aktivitäten während der EdScratch-Lektionen den Schülern die Möglichkeit bieten, mit Edison kreativ zu denken und zu kreieren, bietet diese Aktivität einen expliziten Rahmen für die Annäherung an Physical-Computing-Projekte. Der Hauptzweck dieser Aktivität ist es, den Schaffensprozess auf strukturierte Weise vorzustellen und zu zeigen, wie wertvoll es ist, sich die Zeit zum Planen (Entwerfen) zu nehmen - was ein wichtiger Teil der Programmier- und Ingenieurpraxis ist. Schülerinnen und Schüler machen ein Brainstorming möglicher Ideen für "Erfindungen" mit Edison und arbeiten sich dann durch den Prozess des Entwurfs sowohl der physischen Ingenieurskomponente als auch der Programmierkomponente ihrer gewählten Idee.

Das Erlernen des Design-Build-Test-Zyklus und das Üben der Designkomponente ermöglicht es den Schülern, größere Erfolge in offenen Ingenieur- und Programmierprojekten zu erzielen, einschließlich der anderen Aktivitäten in dieser Lektion. Ein Rahmen für die Herangehensweise an offene Projekte hilft den Schülern, sich auf das Gelernte zu konzentrieren, es zu planen und anzuwenden. Schülerinnen und Schüler mit vielen Ideen, aber unkonzentrierter Energie können durch die Verwendung eines Rahmens ihre Kreativität in den Iterationsprozess einbringen. Bei Schülern, denen es an Selbstvertrauen im Umgang mit offenen Herausforderungen mangelt, kann eine Struktur, auf die sie sich beziehen können, sie ermutigen, neue Dinge auszuprobieren und ihr kreatives Potenzial freizusetzen.

Tipps und Tricks

- Wenn Sie diese Aktivität als Gruppe oder als ganze Klasse durchführen, sollten Sie in Erwägung ziehen, den Timer für die Brainstorming-Sitzung zu verwalten. Wenn Sie die Schüler zwingen, mit ihrer nächsten Idee weiterzumachen, wenn die Zeit abgelaufen ist, können Sie sie auf Trab halten und sie in den Fluss bringen. So können Sie auch alle Schülerinnen und Schüler erkennen, die vielleicht aufgegeben haben, und sie ermutigen, weiterzumachen. Keine schlechten Ideen beim Brainstorming!
- Die anfängliche Brainstorming-Aufgabe lässt sich am besten als Einzelperson erledigen, aber die Überprüfung der Ideen und Entwurfsteile dieser Aktivität kann leicht mit Partnern oder in Gruppen erfolgen.
- Wenn Schülerinnen und Schüler ihr Projekt in mehr als zwei Teile zerlegt haben, lassen Sie sie zusätzliches Papier verwenden oder separate Abschnitte erstellen, um jeden Teil zu entwerfen und zu planen.
- Obwohl die Schülerinnen und Schüler ihre Kreationen eigentlich nicht bauen und programmieren müssen, um die Basisaufgaben zu erledigen, kann sich

der Zugang zu Edison und EdScratch für ihren Designprozess als hilfreich erweisen.

- Der Kern dieser Aktivität durchläuft nur die 'Design'-Phase. Die "Mini-Herausforderung" ist keine Mini-Herausforderung - das tatsächliche Bauen und Testen der von den Schülern erarbeiteten Ideen wird diese Aktivität zu einem großen Projekt machen.
- Ideen von Schülern können möglich oder nicht möglich sein. Wenn Sie Schüler versuchen lassen, ihre Entwürfe zu bauen und zu programmieren, erinnern Sie sie daran, dass, selbst wenn ihr Entwurf nicht realisierbar ist, dies nicht zum Scheitern führt. Schauen Sie sich an, warum es nicht durchführbar ist. Gibt es zum Beispiel eine Grenze für die Funktionsweise des Roboters oder die Materialien, die sie verwendet haben und die den Misserfolg verursacht haben?
 - Erfolgreiche Konstruktionen können genauso viel, wenn nicht sogar mehr lehren als erfolgreiche Konstruktionen.

Frage	Typ	Antwort/Beispielantwort	Hinweis
1	SE	<i>Projektname: EggBeater Bot. Meine Idee ist es, Schneebesen an Edisons Steckdosen anzuschließen und Edison in einen Handmixer zu verwandeln.</i>	
2	SE	<i>Es gibt zwei Hauptteile: den physischen Entwurf und das Programm. Jeder dieser Teile besteht aus zwei Teilen: Physikalischer Entwurf - 1) die Herstellung der Schneebesen und 2) das Anbringen der Schneebesen an Edison. Programmdesign - 1) das Programm in die Lage zu versetzen, zu starten und zu stoppen und 2) das Programm dazu zu bringen, die Schneebesen mit verschiedenen Geschwindigkeiten zu bewegen.</i>	Das Projekt sollte eine physikalische Konstruktions- und Software-Programmierkomponente haben.

3	SE	<p><i>Schneebesens: Ich denke, ich werde für den Hauptschaft des Schneebesens einen Stabbleistift verwenden und dann die Klingen mit geschnittenen Plastikstreifen aus recycelten Flaschen herstellen. Diese werde ich auf den Bleistift kleben. Anbringen der Schneebesens: Ich denke, ich werde hauptsächlich EdCreate Teile verwenden, um die Schneebesens an den Rädern von Edison zu befestigen. Auf diese Weise brauche ich keine separate Möglichkeit, sie auch in der Steckdose zu befestigen.</i></p>	<p>Jede Kombination von Zeichnungen, Diagrammen, schriftlichen Plänen usw. ist akzeptabel, solange der Entwurfsplan der Schüler detailliert ist.</p> <p>Die Antwort/Beispielantwort ist bewusst kurz gehalten - sie soll nur als ein mögliches Teilbeispiel dienen.</p>
4	SE	<p><i>Das Programm sollte Tastendruck-Ereignisse verwenden. Wenn die Dreieckstaste gedrückt wird, sollten die Motoren anfangen, sich vorwärts zu drehen. Wenn der runde Knopf gedrückt wird, sollten sich die Motoren um eine Geschwindigkeit schneller bewegen. Die Motoren sollten sich mit jedem runden Knopfdruck schneller bewegen, bis zu einer Höchstgeschwindigkeit von 5. Jedes Mal, wenn der Dreiecksknopf gedrückt wird, während sich die Motoren bewegen, sollten sie anhalten.</i></p>	<p>Pseudocode oder jede Kombination von Zeichnungen, Diagrammen, schriftlichen Plänen usw. ist akzeptabel, solange der Entwurfsplan der Schüler detailliert ist.</p> <p>Die Antwort/Beispielantwort wurde absichtlich kurz gehalten - sie soll nur als mögliches Teilbeispiel dienen.</p>

Lösungsschlüssel

Aktivität U6-1.1a: Eine imaginäre Kreatur erfinden

Umfang der Tätigkeit	Projekt
Empfehlung	- Beenden Sie die Aktivität U6-1.1 vor dieser Aktivität - Die Aktivitäten U6-1.1a, U6-1.1b, U6-1.1c, U6-1.1d und U6-1.1e bieten alle eine offene Projektidee, die von den Schülern die Anwendung des Design-Build-Test-Zyklus verlangt. Erwägen Sie die Verwendung von mindestens einem dieser Zyklen.
Benötigte Materialien	Basissatz, Arbeitsblatt U6-1.1a, Makerspace-/Handwerkerzubehör, EdCreate-Sätze/LEGO-Steine

Überblick

Offene Projekte, einschließlich dieser Herausforderung, fördern die kreative Problemlösung und geben Schülern die Möglichkeit, MINT-Anwendungen für reale Szenarien zu finden. Als eine von fünf Projektoptionen besteht das Ziel dieser Herausforderung darin, dass Schüler eine imaginäre Kreatur bauen, die sich mit mindestens einem von Edisons Sensoren bewegen und auf die Welt reagieren kann.

Schüler wenden den Design-Build-Test-Zyklus auf diese unbefristete Herausforderung an. Mit einem festgelegten Endziel und nur wenigen festgelegten Erfolgskriterien soll diese Aktivität kreative Problemlösungen durch die sinnvolle Anwendung der erlernten Fähigkeiten fördern. Die Schüler müssen ihre Programmier- und Computerfähigkeiten auf diese Herausforderung anwenden, müssen aber auch mit der physischen Kreation experimentieren, was diesem Projekt eine technische Komponente hinzufügt.

Tipps und Tricks

- Dieses Projekt funktioniert gut in Paaren oder Kleingruppen.
- Erinnern Sie die Schüler daran, dass erfolglose Entwürfe genauso viel, wenn nicht sogar mehr, lehren können als erfolgreiche Entwürfe. Ermutigen Sie die Schülerinnen und Schüler, das, was bei ihrer Kreation nicht funktioniert, durch die Linse des Design-Build-Test-Zyklus zu betrachten. Zerlegen Sie das, was vor sich geht, in kleinere Einheiten und lösen Sie jede Komponente.
- Es gibt keinen "besten Weg", sich dem Problem oder einer vorgegebenen Lösung zu nähern. Die Kreationen der Schüler können sehr unterschiedlich sein, ebenso wie die Programmierkomponente ihrer Lösungen.

Aktivität U6-1.1b: Einen Wattebausch-Werfer erfinden

Umfang der Tätigkeit	Projekt
Empfehlung	- Beenden Sie die Aktivität U6-1.1 vor dieser Aktivität - Die Aktivitäten U6-1.1a, U6-1.1b, U6-1.1c, U6-1.1d und U6-1.1e bieten alle eine offene Projektidee, die von den Schülern die Anwendung des Design-Build-Test-Zyklus verlangt. Erwägen Sie die Verwendung von mindestens einem dieser Zyklen.
Benötigte Materialien	Basissatz, Arbeitsblatt U6-1.1b, Raum- und Bastelbedarf, EdCreate-Sätze/LEGO-Steine, Wattebällchen oder Bastel-Bommeln (zum Abschuss)

Überblick

Offene Projekte, einschließlich dieser Herausforderung, fördern die kreative Problemlösung und geben Schülern die Möglichkeit, MINTAnwendungen für reale Szenarien zu finden. Als eine von fünf Projektoptionen besteht das Ziel dieser Herausforderung darin, dass Schüler einen Wattebausch-Werfer bauen, der so hoch wie möglich, so weit wie möglich oder so genau wie möglich schießen kann.

Schüler wenden den Design-Build-Test-Zyklus auf diese unbefristete Herausforderung an. Mit einem festgelegten Endziel und der Wahl des Ziels als einzigem festgelegten Erfolgskriterium soll diese Aktivität die kreative Problemlösung durch die sinnvolle Anwendung der erlernten Fähigkeiten fördern. Die Schüler müssen ihre Programmier- und Computerefähigkeiten auf diese Herausforderung anwenden, müssen aber auch mit der physischen Kreation experimentieren, was diesem Projekt eine technische Komponente hinzufügt.

Tipps und Tricks

- Dieses Projekt funktioniert gut in Paaren oder Kleingruppen.
- Erinnern Sie die Schüler daran, dass erfolglose Entwürfe genauso viel, wenn nicht sogar mehr, lehren können als erfolgreiche Entwürfe. Ermutigen Sie die Schülerinnen und Schüler, das, was bei ihrer Kreation nicht funktioniert, durch die Linse des Design-Build-Test-Zyklus zu betrachten. Zerlegen Sie das, was vor sich geht, in kleinere Einheiten und lösen Sie jede Komponente.
- Es gibt keinen "besten Weg", sich dem Problem oder einer vorgegebenen Lösung zu nähern. Die Kreationen der Schüler können sehr unterschiedlich sein, ebenso wie die Programmierkomponente ihrer Lösungen.

Aktivität U6-1.1c: Einen Einbrecheralarm erfinden

Umfang der Tätigkeit	Projekt
Empfehlung	- Beenden Sie die Aktivität U6-1.1 vor dieser Aktivität - Die Aktivitäten U6-1.1a, U6-1.1b, U6-1.1c, U6-1.1d und U6-1.1e bieten alle eine offene Projektidee, die von den Schülern die Anwendung des Design-Build-Test-Zyklus verlangt. Erwägen Sie die Verwendung von mindestens einem dieser Zyklen.
Benötigte Materialien	Basissatz, Arbeitsblatt U6-1.1c, Raum- und Bastelmaterial für Maker, EdCreate-Sätze/LEGO-Steine

Überblick

Offene Projekte, einschließlich dieser Herausforderung, fördern die kreative Problemlösung und geben Schülern die Möglichkeit, STEM-Anwendungen für reale Szenarien zu finden. Als eine von fünf Projektoptionen besteht das Ziel dieser Herausforderung darin, dass Schüler eine Einbruchmeldeanlage bauen, die mindestens einen von Edisons Sensoren verwendet, um eine "Bedrohung" zu erkennen und dann eine Ausgabe zu erzeugen, um den potenziellen Dieb abzuschrecken.

Die Schüler wenden bei dieser offenen Herausforderung den Design-Build-Test-Zyklus an. Mit einem festgelegten Endziel und nur wenigen festgelegten Erfolgskriterien soll diese Aktivität durch die sinnvolle Anwendung der erlernten Fähigkeiten zur kreativen Problemlösung anregen. Die Schüler müssen ihre Programmier- und Computerfähigkeiten auf diese Herausforderung anwenden, müssen aber auch mit der physischen Kreation experimentieren, was diesem Projekt eine technische Komponente hinzufügt.

Tipps und Tricks

- Dieses Projekt funktioniert gut in Paaren oder Niedrig-Gruppen.
- Erinnern Sie die Schüler daran, dass erfolglose Entwürfe genauso viel, wenn nicht sogar mehr, lehren können als erfolgreiche Entwürfe. Ermutigen Sie die Schülerinnen und Schüler, das, was bei ihrer Kreation nicht funktioniert, durch die Linse des Design-Build-Test-Zyklus zu betrachten. Zerlegen Sie das, was vor sich geht, in kleinere Einheiten und lösen Sie jede Komponente.

- Es gibt keinen einzigen "besten Weg", sich dem Problem oder einer vorgegebenen Lösung zu nähern. Die Kreationen der Schüler können sehr unterschiedlich sein, ebenso wie die Programmierkomponente ihrer Lösungen.

Activity U6-1.1d : Invent a mousetrap

Umfang der Tätigkeit	Projekt
Empfehlung	- Beenden Sie die Aktivität U6-1.1 vor dieser Aktivität - Die Aktivitäten U6-1.1a, U6-1.1b, U6-1.1c, U6-1.1d und U6-1.1e bieten alle eine offene Projektidee, die von den Schülern die Anwendung des Design-Build-Test-Zyklus verlangt. Erwägen Sie die Verwendung von mindestens einem dieser Zyklen.
Benötigte Materialien	Basissatz, Arbeitsblatt U6-1.1d, Hersteller-Raum-/Handwerkerzubehör, EdCreate-Sätze/LEGO-Steine

Überblick

Offene Projekte, einschließlich dieser Herausforderung, fördern die kreative Problemlösung und geben Schülern die Möglichkeit, STEM-Anwendungen für reale Szenarien zu finden. Als eine von fünf Projektoptionen besteht das Ziel dieser Herausforderung darin, dass Schüler eine Mausefalle bauen, die mindestens einen von Edisons Sensoren verwendet, um die Falle auszulösen und dann eine Ausgabe zu produzieren, die den Schöpfer darauf aufmerksam macht, dass die Falle gesprungen ist.

Die Schüler wenden bei dieser offenen Herausforderung den Design-Build-Test-Zyklus an. Mit einem festgelegten Endziel und nur wenigen festgelegten Erfolgskriterien soll diese Aktivität durch die sinnvolle Anwendung der erlernten Fähigkeiten zur kreativen Problemlösung anregen. Die Schüler müssen ihre Programmier- und Computerfähigkeiten auf diese Herausforderung anwenden, müssen aber auch mit der physischen Kreation experimentieren, was diesem Projekt eine technische Komponente hinzufügt.

Tipps und Tricks

- Dieses Projekt funktioniert gut in Paaren oder Niedrig-Gruppen.
- Erinnern Sie die Schüler daran, dass erfolglose Entwürfe genauso viel, wenn nicht sogar mehr, lehren können als erfolgreiche Entwürfe. Ermutigen Sie die Schülerinnen und Schüler, das, was bei ihrer Kreation nicht funktioniert, durch die Linse des Design-Build-Test-Zyklus zu betrachten. Zerlegen Sie das, was vor sich geht, in kleinere Einheiten und lösen Sie jede Komponente.

- Es gibt keinen einzigen "besten Weg", sich dem Problem oder einer vorgegebenen Lösung zu nähern. Die Kreationen der Schüler können sehr unterschiedlich sein, ebenso wie die Programmierkomponente ihrer Lösungen.

Aktivität U6-1.1e: Einen Kombinationssafe erfinden

Umfang der Tätigkeit	Projekt
Empfehlung	- Beenden Sie die Aktivität U6-1.1 vor dieser Aktivität - Die Aktivitäten U6-1.1a, U6-1.1b, U6-1.1c, U6-1.1d und U6-1.1e bieten alle eine offene Projektidee, die von den Schülern die Anwendung des Design-Build-Test-Zyklus verlangt. Erwägen Sie die Verwendung von mindestens einem dieser Zyklen.
Benötigte Materialien	Basissatz, Arbeitsblatt U6-1.1e, Makerspace/Baumaschinenzubehör, EdCreate-Sätze/LEGO-Steine, TV/DVD-Fernbedienungen (als Option für den Tresor-Code)

Überblick

Offene Projekte, einschließlich dieser Herausforderung, fördern die kreative Problemlösung und geben Schülern die Möglichkeit, MINT-Anwendungen für reale Szenarien zu finden. Als eine von fünf Projektoptionen besteht das Ziel dieser Herausforderung darin, dass Schüler einen Kombinationstresor bauen, der sich nur öffnet, wenn die richtige Sequenz (Code) eingegeben wird. Der Code, der den Tresor öffnet, kann eine Sequenz von Dreiecks- und runden Tastendrücken, eine Sequenz von TV/DVD-Ferncodesignalen, eine Sequenz von IR-Nachrichten von einem anderen Edison-Roboter oder eine Kombination all dieser Optionen sein.

Die Schüler wenden bei dieser offenen Herausforderung den Design-Build-Test-Zyklus an. Mit einem festgelegten Endziel und der Wahl des Ziels als einzigem festgelegtem Erfolgskriterium soll diese Aktivität die kreative Problemlösung durch die sinnvolle Anwendung der erlernten Fähigkeiten fördern. Die Schüler müssen ihre Programmier- und Computerfähigkeiten auf diese Herausforderung anwenden, müssen aber auch mit der physischen Kreation experimentieren, was diesem Projekt eine technische Komponente hinzufügt.

Tipps und Tricks

- Dieses Projekt funktioniert gut in Paaren oder Kleingruppen.
- Erinnern Sie die Schüler daran, dass erfolglose Entwürfe genauso viel, wenn nicht sogar mehr, lehren können als erfolgreiche Entwürfe. Ermutigen Sie die Schülerinnen und Schüler, das, was bei ihrer Kreation nicht funktioniert, durch die Linse des Design-Build-Test-Zyklus zu betrachten. Zerlegen Sie das, was vor sich geht, in kleinere Einheiten und lösen Sie jede Komponente.
- Es gibt keinen "besten Weg", sich dem Problem oder einer vorgegebenen Lösung zu nähern. Die Kreationen der Schüler können sehr unterschiedlich sein, ebenso wie die Programmierkomponente ihrer Lösungen.

Aktivität U6-1.2 Wir erkunden ein Spukhaus

Umfang der Tätigkeit	Projekt
Empfehlung	Als Eckpfeiler des Projekts sowohl für Einheit 6 als auch für die EdScratch-Lektion empfohlen
Benötigte Materialien	Basissatz, Arbeitsblatt U6-1.2, Makerspace/Baumaterial, EdCreate-Sätze/LEGO-Steine, eine schwarze Arbeitsfläche, weißes Abdeckband oder o.Ä.

Überblick

Dieses Projekt erinnert mehr an eine "Change it up"-Aktivität und unterscheidet sich von allen anderen "Let's explore"-Aktivitäten in den Unterrichtsplänen. Diese Aktivität, die sowohl für die Einheit 6 als auch für den gesamten EdScratch-Unterricht als zentrales Projekt konzipiert wurde, fordert die Schüler auf, ihrer Fantasie freien Lauf zu lassen. Indem sie ihre eigenen Szenarien, Testumgebungen, Erfindungen und Programme erstellen, haben die Schülerinnen und Schüler die Möglichkeit, kreative Problemlösungen und ihre Programmier- und Computerfähigkeiten anzuwenden.

Dieses Projekt ist teils als Design-Herausforderung, teils als Programmier-Herausforderung und teils als Übung in Zusammenarbeit konzipiert und soll in Gruppen durchgeführt werden, wobei nur sehr wenige Einschränkungen oder Erfolgsparameter vorgegeben sind. Um den Schülern den Einstieg in die experimentelle Denkweise zu erleichtern, stellt die erste Aufgabe in diesem Projekt (die Geisterjäger-Herausforderung) ein strukturierteres Szenario dar. Was der Raum (der Testraum) sein muss, ist bereits festgelegt, und es gibt Hinweise, was in die Programmierlösung einbezogen werden muss. Außerdem entfällt die Notwendigkeit, eine physische Kreation zu bauen. Sobald die Schüler diesen Raum jedoch gelöst haben, verschwinden die Grenzen.

Programmieren ist von Natur aus kreativ, und ein Großteil der Technologie existiert außerhalb des Computerbildschirms. Bei diesem Projekt beenden die Schüler ihre EdScratch-Erfahrung mit einer Herausforderung, die es ihnen erlaubt, fantasievoll und einfallsreich zu sein und Spaß zu haben. Auf diese Weise festigen die Schüler das Gelernte, stärken ihre Belastbarkeit und Problemlösungskapazität und werden ermutigt, die nächsten Schritte in ihren Programmier- und Robotikabenteuern zu gehen.

Tipps und Tricks

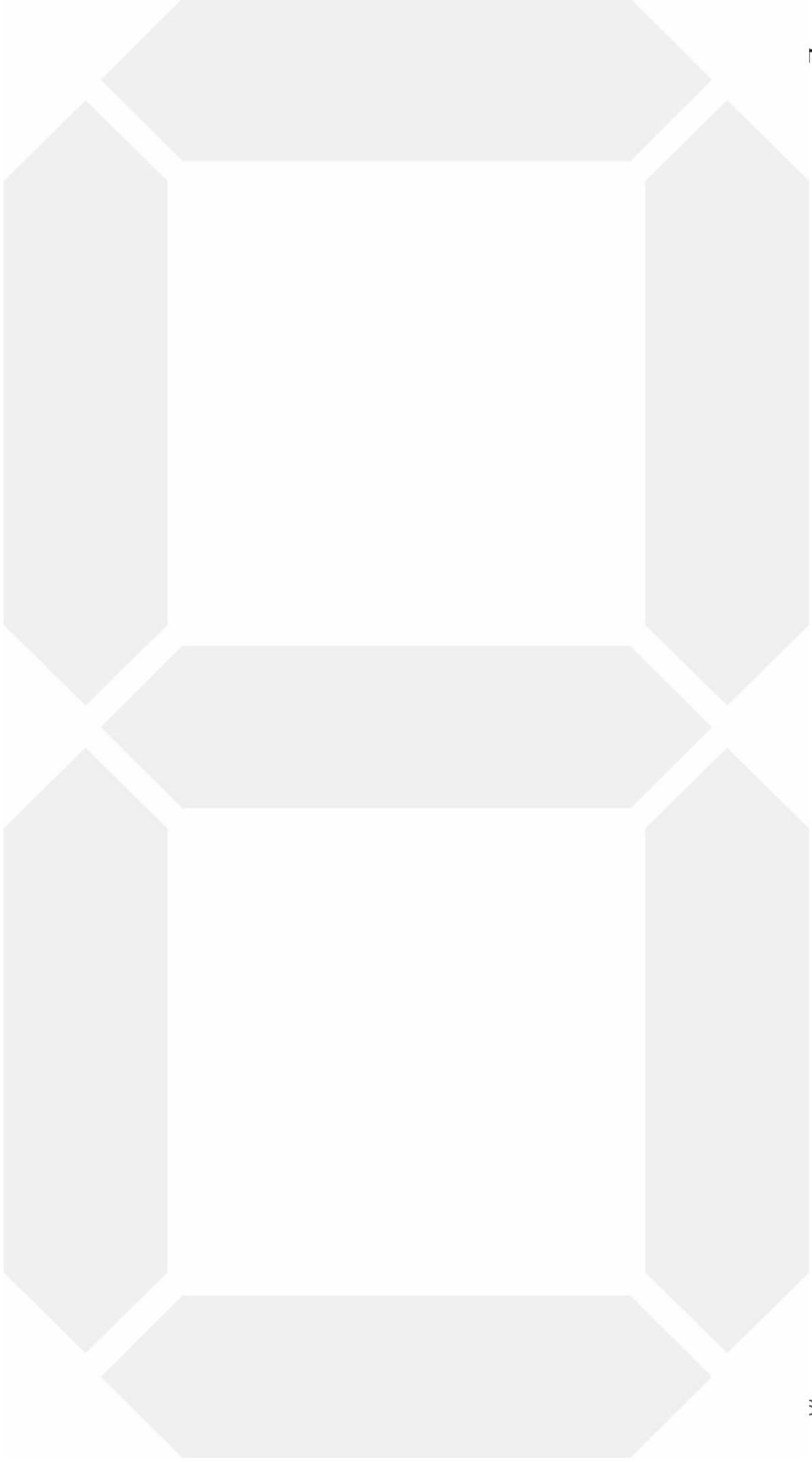
- Dieses Projekt soll in Gruppen durchgeführt werden.

- Eine mögliche Lösung für das Geisterjagdzimmer kann unter <https://www.edscratchapp.com?share=B05j61bZ> eingesehen werden.
- Erinnern Sie die Schüler daran, dass erfolglose Entwürfe genauso viel, wenn nicht sogar mehr als erfolgreiche Entwürfe lehren können. Ermutigen Sie die Schülerinnen und Schüler, das, was bei ihrer Kreation nicht funktioniert, durch die Linse des Design-Build-Test-Zyklus zu betrachten. Zerlegen Sie das, was vor sich geht, in kleinere Einheiten und lösen Sie jede Komponente.

Lösungsschlüssel

Das Arbeitsblatt hat zwar Leerzeichen für "Antworten", aber diese Bereiche müssen nicht markiert werden. Lassen Sie die Schülerinnen und Schüler diese Bereiche als losen Rahmen und Arbeitsbereich nutzen, um ihre Entwürfe zu erfassen. In diesem Projekt gibt es keine "richtigen" Antworten. Ermutigen Sie die Schüler dazu, kreativ zu sein, Risiken einzugehen, das Gelernte anzuwenden und Spaß zu haben!

Anhang 1: Vorlage digitale Nummernschablone



Anhang 2: Kalibrieren der Hinderniserkennung

Sie können die Empfindlichkeit des Hinderniserkennungssystems von Edison regulieren. Indem Sie das Hinderniserkennungssystem empfindlicher machen, kann Edison weiter entfernte Hindernisse erkennen. Indem man das System weniger empfindlich macht, kann Edison nur sehr nahe Hindernisse erkennen. Folgen Sie den Anweisungen auf diesem Blatt, um das Hinderniserkennungssystem Ihres Edison einzustellen.

Lesen Sie den Strichcode

1. Platzieren Sie Edison auf der rechten Seite des Strichcodes, dem Strichcode zugewandt.
2. Drücken Sie dreimal auf den Aufnahmeknopf (rund).
3. Warten Sie, während Edison vorwärts fährt und den Strichcode scannt.



Barcode - Hinderniserkennung kalibrieren

Maximale Empfindlichkeit einstellen

Nachdem Sie den Strichcode gescannt haben, legen Sie Edison auf einen Tisch oder Schreibtisch und entfernen Sie alle Hindernisse vor Edison. Drücken Sie dann die Taste abspielen (Dreieck). Edison befindet sich jetzt im Kalibrierungsmodus.

Zuerst wird die linke Empfindlichkeit kalibriert.

1. Drücken Sie wiederholt die Taste abspielen (Dreieck), was die Empfindlichkeit erhöht, bis die rote LED links flackert.
2. Drücken Sie wiederholt die Aufnahmetaste (rund), wodurch die Empfindlichkeit verringert wird, bis die LED vollständig aufhört zu flackern.
3. Drücken Sie die Stopptaste (quadratisch), um zum Kalibrieren der rechten Seite umzuschalten.
4. Drücken Sie wiederholt die Taste abspielen (Dreieck), bis die rechte rote LED flackert.
5. Drücken Sie wiederholt den Aufnahme-Knopf (rund), bis die LED vollständig aufhört zu flackern.
6. Drücken Sie den Stop-Knopf, um die Kalibrierung abzuschließen.

Besonderer Hinweis: Benutzerdefinierte Empfindlichkeit

Es ist möglich, den Abstand einzustellen, in dem Hindernisse erkannt werden. Scannen Sie dazu den Strichcode "Hinderniserkennung kalibrieren" ein, platzieren Sie ein Hindernis vor Edison in der Entfernung, in der Edison Hindernisse erkennen soll, drücken Sie die Taste abspielen und wiederholen Sie dann die Schritte 1 bis 6, um die Empfindlichkeit einzustellen.